AD-A109 689 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 15/7

AN OPERATIONAL LANCHESTER-TYPE MODEL OF SMALL-UNIT AMPHIBIOUS O--ETC(U)

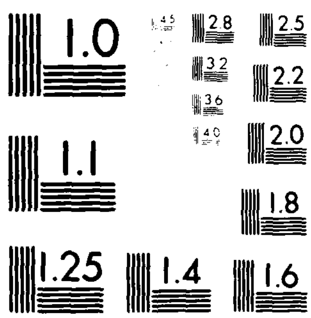
SEP 81 S D PARK

UNCLASSIFIED NL

1 OF 2

AD

A109689



MICROCOPY RESOLUTION TEST CHART

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN OPERATIONAL LANCHESTER-TYPE MODEL OF SMALL-UNIT AMPHIBIOUS OPERATIONS

by

Soon Dae Park

September 1981

Thesis Advisor: J. G. Taylor

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A109 584	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Operational Lanchester-Type Model of Small-Unit Amphibious Operations		5. TYPE OF REPORT & PERIOD COVERED Master's thesis; September 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Soon Dae Park		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 100
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lanchester-type combat model

Amphibious operation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents an operational Lanchester-type model of small-unit amphibious operations. This relatively simple model has been developed to demonstrate the basics of model building to the beginning student interested in amphibious warfare. The model is a time sequenced, deterministic, force-on-force combat model that is implemented on a digital computer. A brief discussion of considerations for modeling amphibious operations is given. The

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#20 - ABSTRACT - (CONTINUED)

details of the model are presented for a specific amphibious-warfare scenario. Additionally, a computer terrain-contour-line plotting program is provided to assist the combat modeler to fit a parameterized-terrain to real terrain.

A

Approved for public release; distribution unlimited.

An Operational Lanchester-Type Model of Small-Unit Amphibious Operations

by

Soon Dae Park
Lieutenant Colonel, Republic of Korea Marine Corps
B.S., Republic of Korean Naval Academy, 1965

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL

September 1981

Author: Park Soon Dae

Approved by: James G. Taylor, Thesis Advisor

Jeffrey L. Ellis, Second Reader

Kneale T. Marshall, Chairman, Department of Operations Research

W. M. Woods, Dean of Information and Policy Sciences

ABSTRACT

This thesis presents an operational Lanchester-type model of small-unit amphibious operations. This relatively simple model has been developed to demonstrate the basics of model building to the beginning student interested in amphibious warfare. The model is a time sequenced, deterministic, force-on-force combat model that is implemented on a digital computer. A brief discussion of considerations for modeling amphibious operations is given. The details of the model are presented for a specific amphibious-warfare scenario. Additionally, a computer terrain-contour-line plotting program is provided to assist the combat modeler to fit a parameterized-terrain to real terrain.

TABLE OF CONTENTS

I. INTRODUCTION

During the past several decades, combat models have been widely used to support military decisions. As the art of combat modeling becomes more advanced, combat modelers are continuously building more and more complicated models. To the beginning modeler, the ability to understand how those models operate is difficult, if not impossible. It is the purpose of this thesis to develop a simple amphibious-operations model that will demonstrate the basics of model building to the beginning student interested in amphibious warfare. In the broadest sense, an amphibious operation is a combined-arms operation which includes all forms of combat--land, air, sea. This thesis will limit itself to the small-unit amphibious operation.

This project started with two basic models: one was developed as the auxiliary model for the evaluation of design and employment alternatives for the LVA (Landing Vehicle Assault) in the thesis of David Larkin Chadwick (September 1978). The other is the Smoler-Mills model which was developed in the thesis of Josef Smoler (September 1979) and enriched in the thesis of Glen Mills (September 1980).

In Chapter II, general considerations for modeling amphibious operations are briefly discussed. Then, a small-scale computer-based Lanchester-type amphibious-operations model is presented, including analytical details of the algorithms used

to represent each of the combat processes considered. Although the model was developed for a specific scenario, it is sufficiently general in design so that it can be adapted to other small-scale amphibious-operations scenarios with only relatively few modifications.

II. GENERAL CONSIDERATIONS FOR MODELING AMPHIBIOUS OPERATIONS

A. CHARACTERISTICS OF AMPHIBIOUS OPERATIONS

In order to model an amphibious operation, it is first of all necessary to understand what is going on in a real amphibious operation. Only after one knows the details of what is happening in such a complex combat operation, can one begin to sift out the cluttering details and make valid simplifying assumptions to come up with a tractable model.

One of the key characteristics that serves to distinguish amphibious operations from other types of military operations is that a complete military force must be transferred ashore in an orderly manner under the constant pressure of actual or potential attack from hostile forces. Because the over-all amphibious assault requires precise and timely execution, the various component operations must be carried out in a planned sequence (especially early in the assault) according to a strict schedule. This sequence and these schedules, however, must be sufficiently flexible to permit rapid changes in line with unexpected development afloat and ashore.

The notable success of amphibious operations during and after World War II is testimony to the fact that, with proper planning and organization, this dual problem can be solved. In today's environment, it is well known that the modern battlefield will be dominated by highly lethal weapons. This has raised serious questions about the survivability of amphibious

forces. On the other hand, the long-range, high-speed assault potentially gives one the capability to launch assaults from far out to sea, land at times and places of one's choosing, and carry more firepower to accomplish the amphibious assault with greater safety for ships and men. Use of some type of combat model is the only way to explore such issues today. In order to build such a combat model of an amphibious operation, it is necessary to develop and consider detailed and specific information on individual tactical and support elements of the landing force, on the size, numbers, and characteristics of the equipments of these elements, and on the sequence of movement of these elements.

The amphibious operation is a combined operation, the entire spectrum of activities involved in an amphibious operation includes:

- pre-assault bombardment by ships and aircraft
- sea mine clearance
- attack on ships by enemy aircraft and cruise missile
- ship-to-shore movement
- surface assault landing
- helicopter operation
- ground combat between maneuver units
- artillery and naval gunfire support
- tactical aircraft support
- mine warfare (sea and land).

While an amphibious operation is one of the most complex of all military operations, defending against it is even more

complex: it is absolutely impossible for an enemy to defend all coastal areas at all times. The flexibility to conduct helicopter-borne vertical assault and surface-borne assault simultaneously will grealy enhance the complexity of defending against an amphibious assault.

Airborne troops and supplies were valuable during the Second War, and further developments in that direction are under way. But whether or not modern airborne tactics and techniques have supplanted (in a practical sense) seaborne assaults (such as those used from 1942 through 1945 in the Pacific and elsewhere), it should be noted that the military problem of landing forces on shores held by an enemy remains. The emphasis in the future will most likely continue to be on having the ability to project forces from the sea onto a hostile shore and to hold such a beachhead.

B. MODELING APPROACH

All models of military operations must abstract from the real world. Since it is obvious that an engagement between modern military forces is a very complicated process, one has to abstract, aggregate, and interpolate in order to scale a combat process down to manageable size for modeling purposes. A variety of modeling approaches are available. These range from simple Lanchester-type models to highly complicated, computer assisted, high-resolution simulations in which the actions of each individual combatant are traced through a combat scenario second by second. Between these two extremes are

other approaches covering the whole spectrum of land combat, from one-on-one duels to theater-level models covering huge geographical dimensions.

There are basically four different types of combat models: war games, analytical (or mathematical), simulations and some combination of these first three types. According to Bonder [Ref. 2], war games are not a feasible mechanism for analyzing a broad spectrum of system alternatives in a responsive manner to meet a planning cycle requirement. However, they are diagnostic in the sense that they reveal problems that need to be resolved with future systems, and are a viable mechanism for training decision makers. Analytical models seek to describe the combat process mathematically. They simplify the conduct of sensitivity analysis and provide an increased ease in interpreting results, since the dynamics of the combat process are contained in readily examined equations. Analytical models of any degree of complexity usually do not yield convenient analytical solutions but require numerical approximation methods. Simulation is the most widely used technique in military system analysis. Simulation can generally produce very useful data, which are needed for further analysis, and sometimes for planning itself. However, the large amount of detail contained in most Monte Carlo simulations makes it difficult to use as the sole vehicle to single out those systems capabilities, tactics, and environmental conditions which significantly contribute to or delimit the system's effectiveness. Since, as we have seen above, no one type of combat model

is unconditionally preferred to another, it is proposed that a combat model should be selected or designed based on a specific scenario and upon analytical requirements.

In most cases, detailed models are more credible to decision makers. However, for many people such detailed models of large-scale combat operations are far too complicated to be understood, require too much input data, and (in general) are not responsive enough. When one looks at computer storage and run time requirements for even the smallest high resolution model, it is easy to see why a high resolution model of a corps or theater is presently impractical, and is likely to remain so. In order to avoid the complexity of the large-scale model and to better understand land combat there is a growing trend among analysts to combine small unit and large unit models in such a way that the output data of a high-resolution small unit combat model is used as the input data for a low-resolution large unit model. The obvious drawback of this hierarchical-modeling approach is that any errors in the small unit models will be carried through, and possibly multiplied, as the process proceeds from model to model. In the large units the emphasis has been away from simulation and towards detailed Lanchester-based models.

So far the emphasis has been adding more and more detail to the high resolution models so as to pick up as many interactions as possible. No matter how much detail is added to the small-unit simulation, it seems impossible that reality will ever be matched exactly. With this in mind, it is proposed

that a well-constructed Lanchester-based model of small unit engagements could give results that are just as valid as the results of a high-resolution simulation.

III. THE MODEL

A. GENERAL

1. The Scenario

The scenario considers an amphibious-landing team, consisting of reconnaissance, a light infantry unit, and landing-assault vehicles. This team is part of an Amphibious Task Force (ATF), and it disembarks from ships that are on station over the horizon from the selected landing site. The assault vehicles, after transmitting from the amphibious shipping to the designated area for the landing formation, form into conventional landing waves at a distance offshore which is greater than the effective range of the direct-fire weapon systems of the shore-defense force. During the ship-to-shore movement the defender's anti-tank guided missile and improved gunnery system respond to the landing. Naval gun-fire ships provide fire support for the assault team during the ship-to-shore movement and the initial stages of landing.

As the assault vehicles reach the beach, they (together with the assault vehicles and any weapons landed by landing vehicles) launch an attack on the enemy shore defense positions. The defenders occupying those positions fight until their losses exceed a maximum permissible amount. The attacking force, however, continues the assault irregardless of losses incurred. Once the shore assault has been completed, the landing force with tactical mobility moves inland to

carry out the tasks, while the enemy prepares to mount a counter attack.

The attacker may engage the advance force of the defender's initial counter attack force on the way to move inland. The advantage will likely go to that force which has gained the initiative (i.e., the landing force) provided it can maintain its momentum.

2. General Description of the Model

The model developed in this thesis is a time sequenced, deterministic, force-on-force computerized model, coded in FORTRAN. The model conducts the battle in uniform time steps of 10 seconds each. Figure 1 shows the general scheme for the sequence of events in the model. The model simulates two main phases in the amphibious operations: (I) the amphibious-assault phase, and (II) the subsequent ground attack. The framework and the logical interrelationships of these two phases will be discussed in the following subsections.

B. THE AMPHIBIOUS ASSAULT PHASE

1. General

In this phase the model considers attrition between the shore-defense force and the landing-assault force during its water-borne movement and subsequent assault to shore. The model aggregates the various actual combat organizations involved in the waterborne phase of the amphibious operation into several homogeneous combat units. Each of these units is characterized by certain relative offensive and defensive

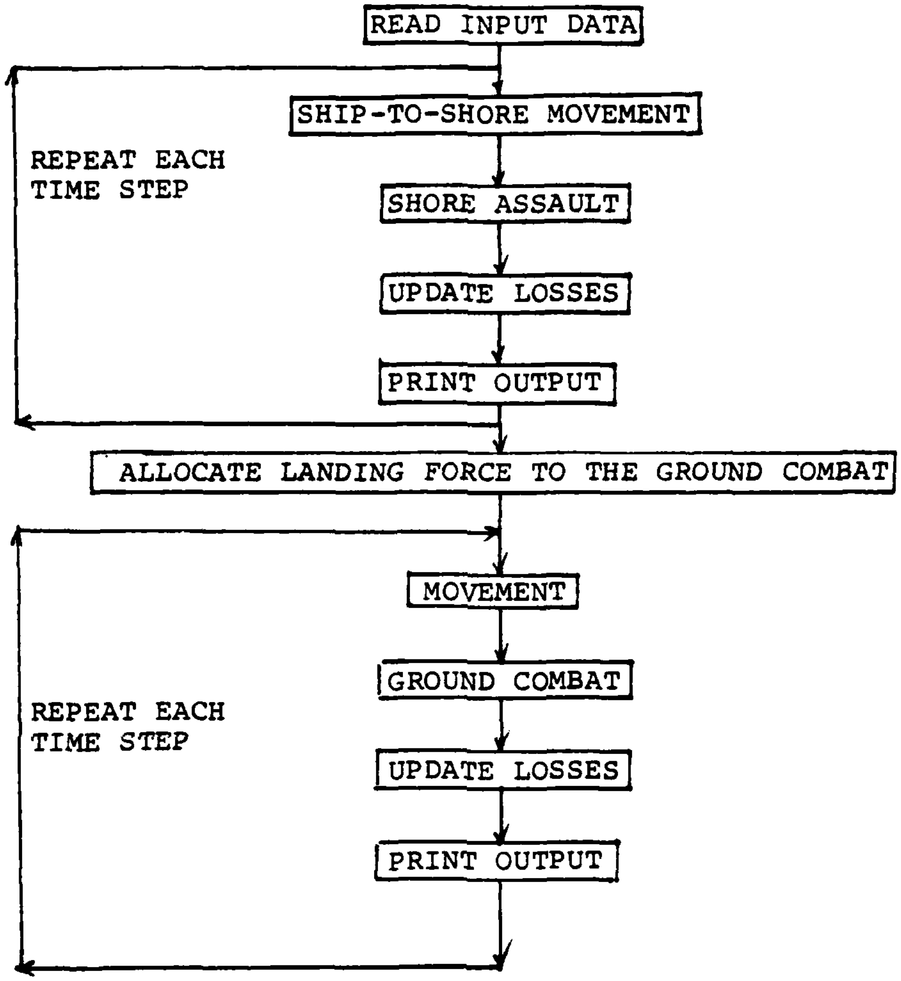


Figure 1. General Scheme for the Amphibious Operation Model

capabilities. The following table illustrates the combat organizations that were explicitly modeled. The combat strength of each unit was represented by the state variables indicated.

combat organization	state variable
Shore Defense--TANK assets	DT
Shore Defense--ATGM assets	DS
Incoming assault waves of LVA representing waves 1 through 5	WV(I), I = 1,2,3,4,5
A cumulative combat force comprised of those Marine ground units which have arrived at the beach and have debarked the LVA	TLF
Fire Support Assets of The Amphibious Task Force	ATFFS

The initial strength in each of the above units is input data to the model.

The schematic of the method of employment for the LVA in the ship-to-shore phase of an amphibious assault is shown in Figure 2. It is assumed that the conventional landing formation composed of waves of landing vehicles will be used as prescribed by current doctrine. The movement of assault vehicles to the beach is simulated using a time step approach. At each time increment the positions of vehicles are updated.

The tactical interrelationships which exist between various combat units are illustrated in Figure 3. Assuming that in such a future amphibious operation the attrition of

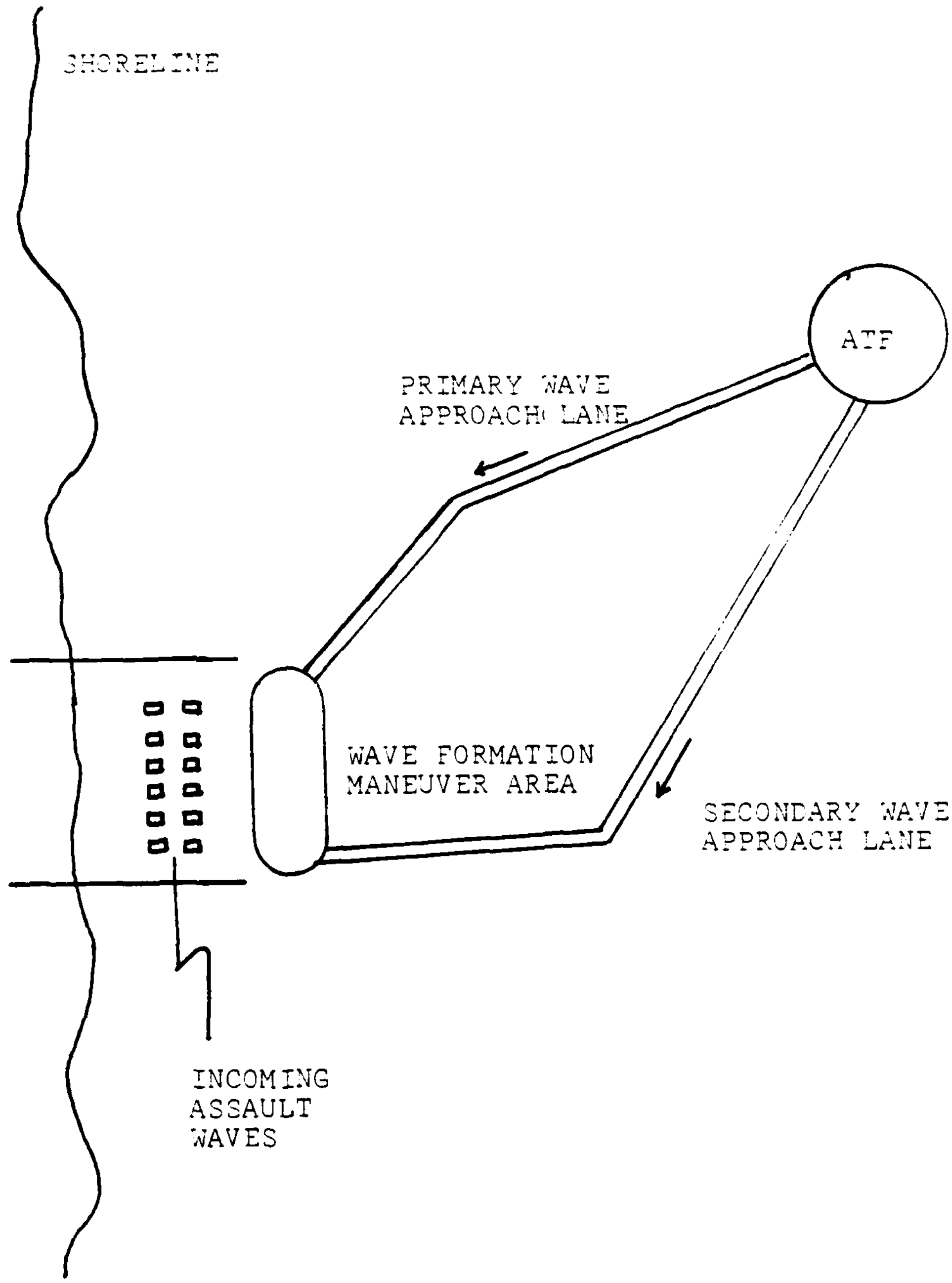


Figure 2. Concept of Ship to Shore Movement

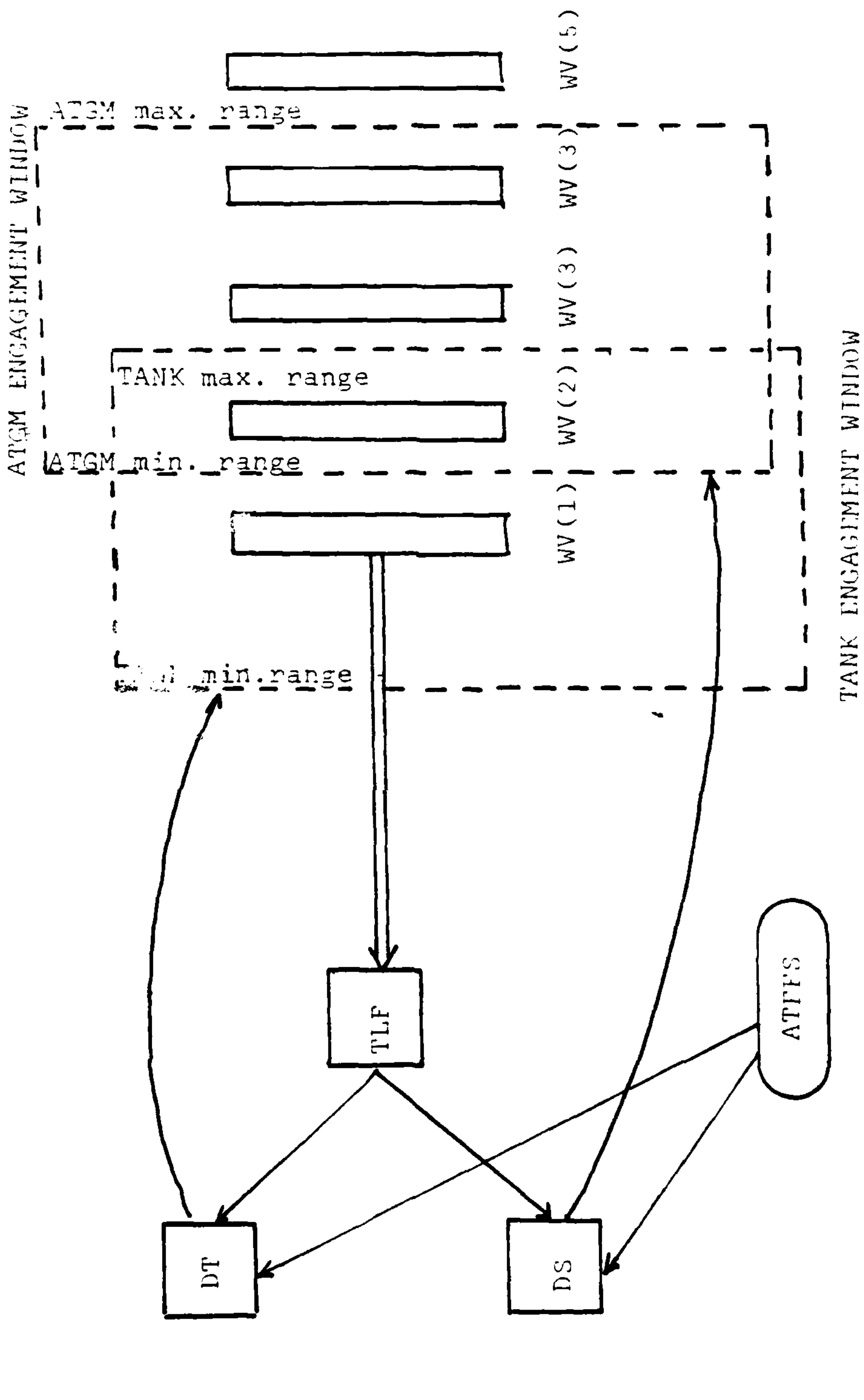


Figure 3. Schematic of Tactical Interrelationship between Combat Units during Amphibious Assault Phase

incoming landing vehicles would be dominated by the effects of shore defense direct-fire weapon systems (specifically, tank and anti-tank guided missile (ATGM) assets), the model essentially omits the effect of the defensive indirect fire capabilities.

2. Attrition Process

The model represents the attrition of all combatant units as a deterministic process. The primary consideration in the ship-to-shore movement of incoming waves of assault vehicle is the attrition effects upon those waves due to the direct fire weapon assets ashore. The attrition of each wave utilizes Lanchester "aimed-fire" equations with variable attrition-rate coefficients.

The classical Lanchester hypothesis for aimed-fire attrition (combat under "modern condition") is that the casualty rate of a unit is proportional to the "size" of the opposing forces. If the unit "A" is being engaged by "D", this may be expressed by the differential equation:

$$\frac{dA}{dt} = - BETA_{DA} \times D$$

The proportionality constant $BETA_{DA}$ is called the Lanchester attrition-rate coefficient. It is assumed that this functional relationship holds for each (firing unit, target unit) pairing over a small time interval dt. The problem then is to determine numerical values for the Lanchester attrition-rate coefficients. In this model, these coefficients were expressed

as the product of the rate of fire (ROF) and the kill probability per round (P_k). Thus,

$$BETA_{DA} = ROF_{DA} \times P(k)_{DA}$$

The rate of fire (ROF) can be expressed as the reciprocal of TBF (Time Between Firings) which can be evaluated by

$$TBF = \text{AIM-RELOAD TIME} + \frac{\text{TARGET RANGE}}{\text{TARGET SPEED + PROJECTIVE VELOCITY}}$$

In determining the probability of a vehicle "KILL" per round, it is assumed that a hit by a large caliber projectile would constitute a "KILL" and the two defensive weapon systems addressed would exhibit normal, uncorrelated horizontal and vertical errors. Then the single shot kill probability is given by

$$P(k) = [(\frac{1}{\sqrt{2\pi}}) \int_{(-a-u)/\sigma_x}^{a-u/\sigma_x} \exp(-\frac{x^2}{2})dx]$$

$$\cdot [(\frac{1}{\sqrt{2\pi}}) \int_{(-b-v)/\sigma_y}^{(b-v)/\sigma_y} \exp(-\frac{y^2}{2})dy]$$

where:

a = semilength of a target

b = semiwidth of a target

u = horizontal aiming error

v = vertical aiming error

σ_x = round-to-round standard deviation in vertical

σ_y = round-tround standard deviation in horizontal

Model functions RNG, HT and SPD are called upon within the model logic to generate the range, height and speed respectively for each assault wave as time is incremented throughout the course of the amphibious assault phase. This information and typical dispersion data (both mean and standard deviation for the tank and ATGM weapons) are then incorporated into the rate of fire and hit probability calculations.

The Amphibious Task Force's fire support assets contribute significantly to the combat effectiveness of the shore defense units. Since it is assumed that the exact positions of the defensive units DT and DS emplaced on shore are unknown to the Amphibious Task Force and consequently the ATF fires into the general areas thought to contain the defensive units. The following Lanchester-type area-fire equations are applied to compute the attrition of DT and DS.

$$\frac{dDT}{dt} = -(ALPHA_{DT} \times ATFFS) \times DT$$

$$\frac{dDS}{dt} = -(ALPHA_{DS} \times ATFFS) \times DS$$

The combat effectiveness of the ATF fire support assets is to be considered relatively constant during this segment of combat time. Thus the terms in parentheses on the right hand side of these equations are to be considered an input parameter.

Once a particular defensive unit has initiated its engagement of incoming waves it is considered that their fire "gives away" their exact locations. At this point it is assumed that the ATF fires will engage that defensive unit through the use of aimed-fire and the loss rate will be in accordance with the Lanchester hypothesis for aimed fire. That is,

$$\frac{dDT}{dt} = -BETA_{DT} \times ATFFS$$

$$\frac{dDT}{dt} = -BETA_{DS} \times ATFFS$$

Again, the parameters on the right-hand sides of both these equations are provided as input.

The casualty rates applied against the DT and DS by the Total Landed Force (TLF) are determined by means of the Lanchester aimed-fire attrition rate coefficients by the equations

$$\frac{dDT}{dt} = -WBETA_{TLF-DT} \times TLF_{DT}$$

$$\frac{dDS}{dt} = -WBETA_{TLF-DS} \times TLF_{DS}$$

The computation of these WBETA coefficients is not performed within the model utilizing the detailed rate of fire and P(k) arguments described previously but is required as input. Although the defensive losses are considered significant,

a high level of complexity for computing these coefficients has not been incorporated into the model at this time.

Figure 4 describes the schematic of the attrition process of the amphibious assault phase in the model. The attrition during each time step was computed using the Euler integration method to approximate Lanchester's force-on-force attrition differential equations.

3. Fire Allocation

Each weapon category was assigned an engagement window as illustrated in Figure 3. Only those LVA located within these range windows could be fired upon by the shore defenders. A defensive weapon only engages the two closest incoming waves if more than two waves of LVA are at any time located within the weapon's engagement window. If only one wave of LVA is present in a weapon's engagement window, defensive fires of that particular weapon type will be distributed uniformly against the surviving LVA in that wave.

If two waves of LVA are both contained within the engagement window, defensive fires of that particular weapon type will be distributed according to a tactical allocation submodel. A weighting factor (DEFWT) is utilized in establishing the proportion of the total weapon strength to be allocated against the surviving LVAs in each of the two waves. As an example, if DEFWT(1) = 2 and DEFWT(2) = 1, then each surviving LVA in the closer of the two incoming waves would be allocated twice as much fire as surviving LVA in the seaward

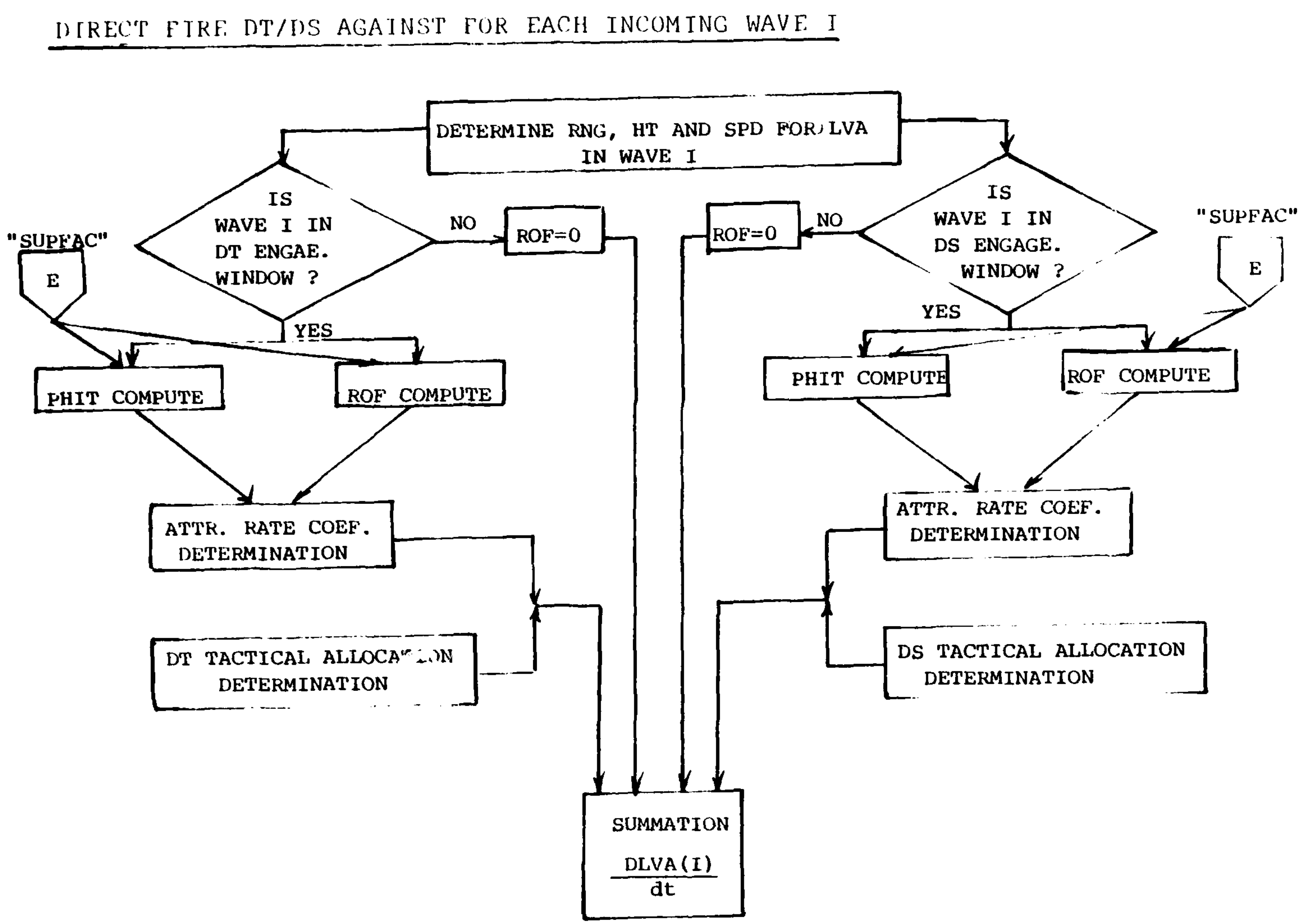


Figure 4. Schematic of The Attrition Pocess for The Assault Phase

ATTRITION FOR THE SHORE DEFENSE FORCE

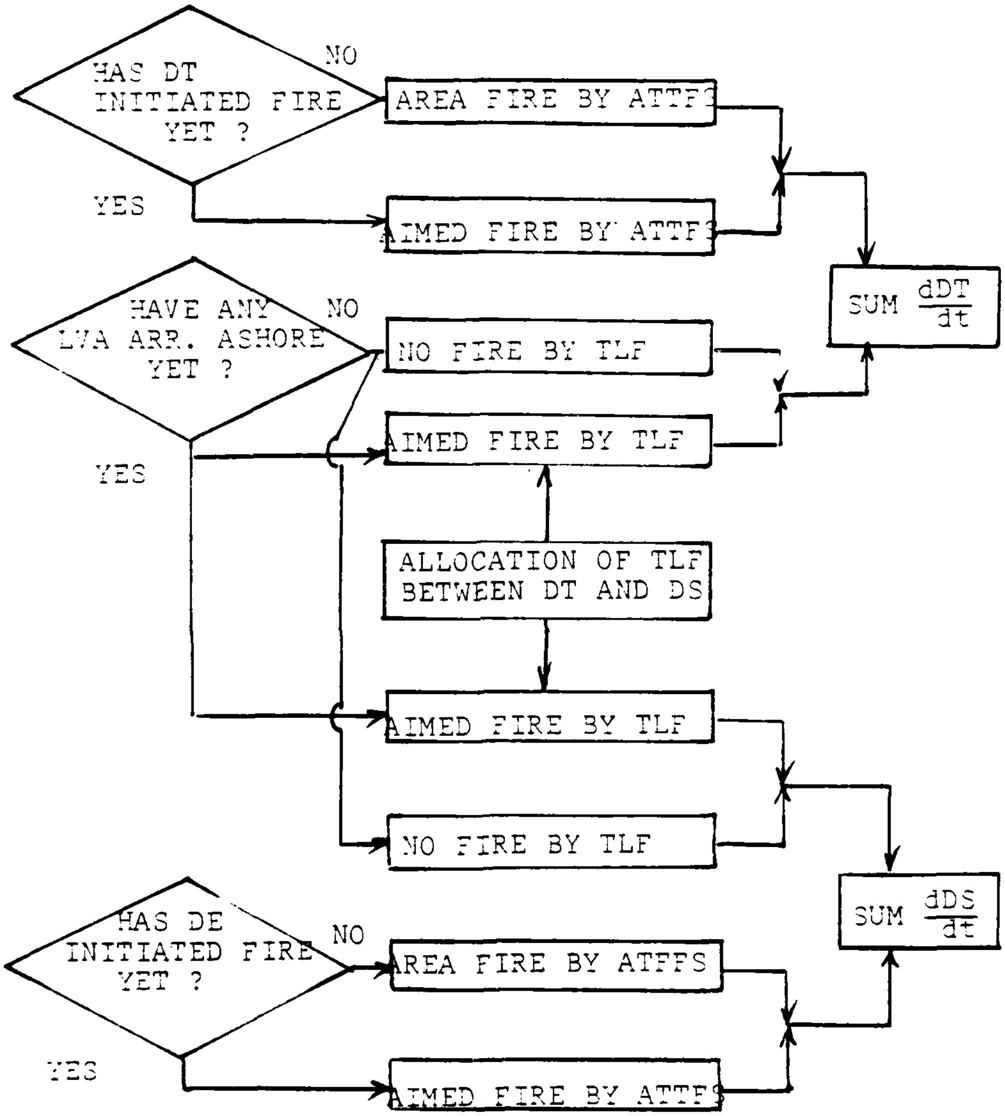


Figure 4.(cont)

wave. For the purpose of these examples, if waves 3 and 4 were both located within the tank engagement window, then the proportion of DT's fire allocated to surviving LVA in wave 3 would be

$$\frac{\text{DEFWT}(1) \times \text{WV}(3)}{\text{DEFWT}(1) \times \text{WV}(3) + \text{DEFWT}(2) \times \text{WV}(4)} \times \text{DT}$$

where WV(3) is the state variable for the current number of survivors in wave 3.

As each assault wave arrives at the beach, the total surviving strength of that wave is transferred to the variable TLF (Total Landed Force). TLF represents a ground combat force equal to that transported by the number of LVA survivors having arrived ashore. Once established, TLF engages the two shore defensive units allocating its fires between the two defensive weapon categories in the same proportion as the number of surviving tanks and ATGM's, that is

$$\text{TLF}_{\text{DS}} = \frac{\text{DS}}{\text{DT} + \text{DS}} \times \text{TLF}$$

$$\text{TLF}_{\text{DT}} = \frac{\text{DT}}{\text{DT} + \text{DS}} \times \text{TLF}$$

4. Suppression

The suppression effects of incoming fire upon each of the defensive units was considered a significant factor with respect to its effect on the survivability of the incoming assault waves of LVA. Generally, the effect of suppression

fire will hinder an individual from observing and firing at the enemy.

It was assumed that suppression would degrade unit effectiveness by increasing the aim-reload time (ARTM) and round-to-round error standard deviation for each weapon system. Hypothesizing that attrition rate is the dominating variable, and therefore, a good indicator of the suppression level, ARTM and such round-to-round errors were assumed to be functions of the force's attrition rate. This is an area, however, requiring further study. Analytically,

$$ARTM_{sup} = ARTM_{nonsup}(1 + GAMMA \times DA)$$

$$ERROR_{sup} = ERROR_{nonsup}(1 + DELTA \times DA)$$

where:

DA	=	attrition rate for defensive unit due to the effect of AFTTS and TLF
GAMMA	=	parameter representing relationship between DA and ARTM
DELTA	=	parameter representing relationship between DA and error standard deviation

This increase of ARTM and round-to-round error (expressed as a standard deviation) decreases the kill probability (P_k) for both defensive weapon categories. Parameter estimation would appear to be the largest problem. But, since determining these parameters in the model is beyond the scope of this thesis, these parameters GAMMA, DELTA are provided as input.

5. The Termination of the Assault Phase

It is assumed that if during the course of the amphibious operation the shore defense forces suffer a cumulative loss in excess of 70% of their initial force strength, the remaining shore defense will try to withdraw, resulting in termination of the engagement.

C. THE INITIATION OF GROUND ATTACK

In the amphibious operation, the landing-force must sieze critically-important inland objectives as rapidly as possible before the defenders start to react to the landing. The decision for the initiation of ground attack should be based upon the enemy threat and desired landing-force build-up ashore, among other factors. To model this decision rule, it is assumed that once the landing has begun, the landing-force commander will base his decision about initiation of ground attack primarily on the strength of the landing-froce ashore and the shore defender's strength. The criteria for the decision should meet these two conditions:

(1) The survived landing-force strength has to be greater than the minimum force required to carry out the ground attack.

(2) The defender's strength must fall below the minimum required to continue coordinated shore defense before breakoff and retreating.

These conditions are then checked after each time step. If all waves landed without reaching the above second conditions, it

is assumed that the next wave group will engage any leftover defenders. Thus, the decision to implement the ground attack is based on the size of the total landed force.

D. *THE GROUND ATTACK PHASE*

The attacking force which is composed of three subunits of three LVAs armed with TOW antitank missile system attacks along predetermined routes. The defending force is comprised of three subunits of three tanks in a static defense.

The battle takes place on parameterized terrain which will be discussed later. The ground-attack process contains five main subprocesses: (1) movement, (2) detection, (3) fire-allocation, (4) attrition and (5) battle termination. The general flow of the ground attack phase is shown in Figure 5.

1. Movement Process

Every attack unit is advanced to the next interval along a predetermined route unless this unit is destroyed already or is in firing status. To use his own determined routes, the user is required to input the original location of each attacking subunit and from one to ten nodes he wishes each attacking subunit to move through. This information, along with vehicle speed, is used to calculate route intervals that move the attacking unit through each of the designated nodes. The straight line ground distance between the first two adjacent nodes, DIST, is calculated as

$$\text{DIST} = \sqrt{X^2 + Y^2}$$

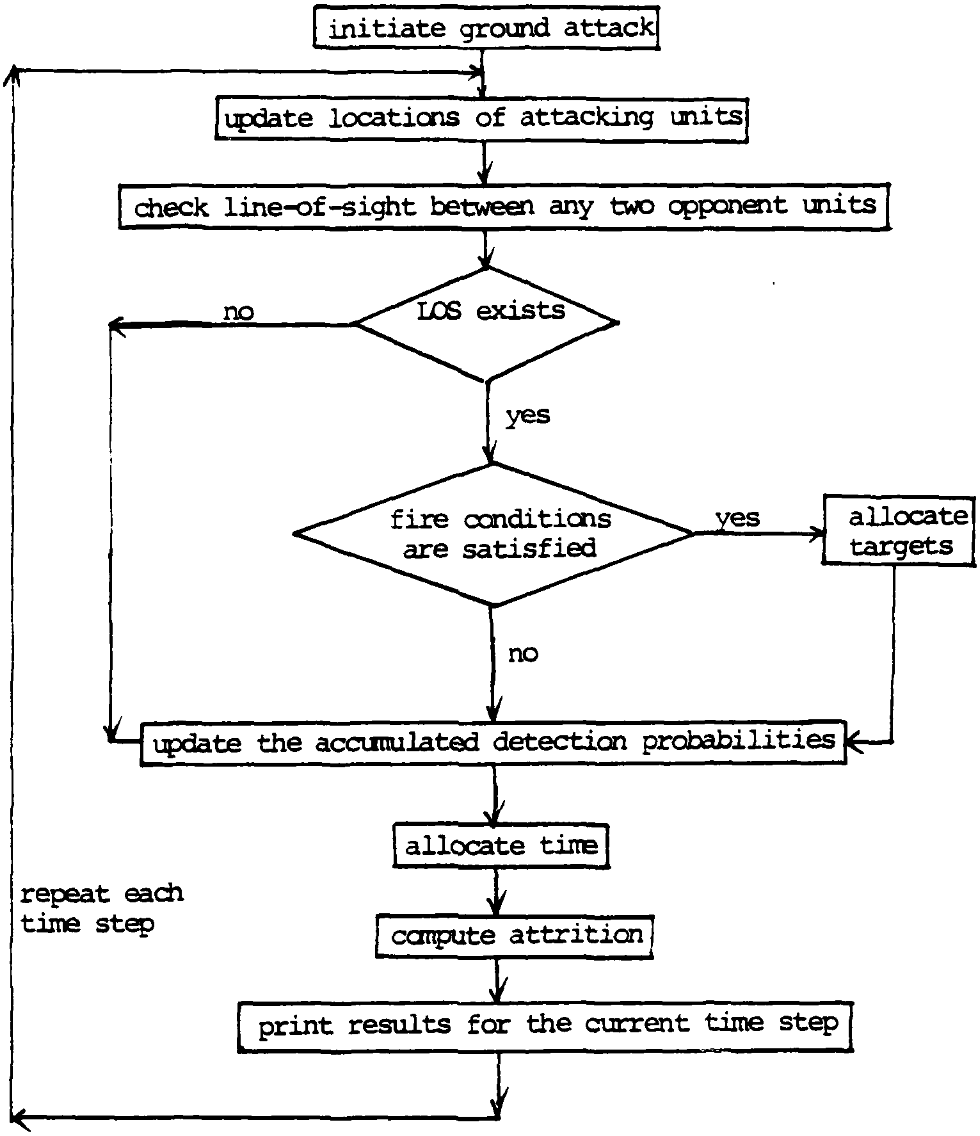


Figure 5. Flow Chart for the Ground Attack Phase

where:

X = distance between two nodes in straight west-east direction

Y = distance between two nodes in straight south to north direction

The angle between the desired direction of movement and straight west to east movement, a, is then calculated. Utilizing these quantities, the distance desired to move during each time step (DST) and the distance to be moved in the X and Y direction (XLN,YLN) is computed. These distances (XLN,YLN) are then added to the coordinates of the previous interval endpoint to determine the coordinates of the next interval endpoint. This process is repeated between the next two nodes until the unit has traversed the entire route.

2. <u>Detection and Fire Allocation</u>

The detection phenomena are modeled in two ways: (I) non-firing detection, and (II) firing-detection. A non-firing detection can occur as a result of an observer's random search within his designated sector of responsibility. The probability (P_k) that an observer A looking at the direction which enables him to detect a target is computed by integrating the Limicon Function over limits are $\pm 15°$ from the primary direction the observer is looking. The Limicon Function, $f(\theta)$, is the following probability density function:

$$f(\theta) = A + B \times \cos(\theta) \qquad -0 \leq \theta \leq 0$$

where:

D = assigned sector width/2

A = $- B \times \cos(D)$

B = $\frac{1}{2}(\sin(D) - D \times \cos(D))$

θ = primary direction observer is looking

Assuming 30° field of view for any observer A target B might be seen only if the observer A is looking at the direction such that ANGRT $\leq \theta \leq$ ANGLFT **where:**

ANGLE = the absolute value of the angle between the primary direction (IPRDIR) and the observer-target direction (OTANG)

ANGLFT = angle + 15° if angle + 15 $\leq$ D

D if angle + 15 > D

ANGRT = angle - 15°

Thus

$$P_k = pr(\text{ANGRT} \leq \alpha \leq \text{ANGLFT})$$

$$= \int_{\text{ANGRT}}^{\text{ANGLET}} f(\alpha)\,d\alpha$$

Given that observer A is looking at the direction, the conditional detection rate (λ_{AB}) is determined by the regression curve [Ref. 11]. The probability that unit j is detected by unit i at time t+Δt[$P_{ij}(t + \Delta t)$] is computed according to:

$$P_{ij}(t + \Delta t) = 1 - e^{-\int_t^{t+\Delta t} \lambda(t)\,dt}$$

$P_{ij}(t)$ can be interpreted as the average fraction of unit i that detects unit j.

The second method of detection played in the model is a so-called firing detection. This phenomena occurs when the following happens: if a firing location is within ±15° of an observer's primary direction of observation when he is firing, he is assumed to be detected and is added to the observer's target list. In summary, the following conditions are necessary for unit j to be a target of unit i:

(a) Line-of-sight must exist between unit i and unit j.

(b) The range between the two units should be between maximum range and minimum range of the firer's weapon system.

(c) $P_{ij}(t - \Delta t) > 0$.

The fire-allocation routine determines what fraction of each unit is allocated to fire each target in target list since it is assumed that each fire unit is not restricted to fire at one target. This fraction is determined as a function of the predetermined fire policy and $P_{ij}(t)$. The fire policy is as follows:

# of target	% of unit i allocated to each target		
	1^{st} priority	2^{nd} priority	3^{rd} priority
1	100%		
2	80%	20%	
3	80%	15%	5%

The priority of a target is taken to be a function of range only. The fire allocation rule which is used in the model is documented in detail in Smoller's thesis [Ref. 9].

3. Attrition

The attrition process in the ground attack phase utilizes Lanchester "aimed-fire" equation used with variable attrition coefficients. The calculation of the attrition coefficients is accomplished through the use of one of two optional methods. The first option uses the following Bonder-Farrell formula to compute the reciprocal of the expected time to kill. The coefficients, A_{ij}, the rate at which one firer of unit i kills unit j targets are computed according to:

$$A_{ij} = 1/E(T_{ij})$$

where $E(T_{ij})$ is the expected time for one firer of unit i to kill one target of unit j. The $E(T_{ij})$ is computed using:

$$E(T_{ij}) = t_a + t_1 - t_h + (t_h + t_f)/P(K/H)$$
$$+ (t_m + t_f)/P(h/m) \times ((1-P(h/h))/P(K/H)$$
$$+ P(h/h) - p)$$

where:

t_a = time to acquire a target

t_1 = time to fire first round following acquisition

t_h = time to fire following a hit

t_m = time to fire following a miss

t_f = time of flight of a round

P = probability of a first round hit

P(h/h) = probability of a hit following a hit

P(h/m) = probability of a hit following a miss

P(k/h) = probability of a kill given a hit

This formula holds for the conditions that the hit probability of any round depends only on the result of the previous round and no accumulated damage is considered. It is assumed that P(K/H) = 1.0 and P(h/m) = p(h/h) = P, thus reducing the equation to:

$$E(T_{ij}) = t_a + t_1 + t_f + (t_m + t_f)(1-P)/P$$

The second method, called the stochastic method, interprets the attrition rate coefficient, A^0_{ij}, as a measure of the fighting ability of a unit which has a random phenomena affected by many different factors. It is assumed that the random fighting ability should be distributed between .3 and .8 with the majority of the unit being rated between .5 and .6. A "fitted" distribution to these assumed fighting levels which is devised by Mills is:

$$A^0_{ij} = -2U^2 + 2U + .3 \text{ ; } U \text{ is a random Uniform } (0,1) \text{ number}$$

The A^0_{ij}'s are a realization of the random variable denoting a unit's initial fighting capability prior to the battle. Then,

during each time step, a new attrition rate coefficient for each unit is computed using the equation:

$$A_{ij} = \begin{cases} A^0_{ij}(1 - r/r_e) \; ; & \text{for } 0 \leq r \leq r_e \\ 0 & \text{for } r_e \leq r \end{cases}$$

where:

r_e = maximum effective range of a firer's weapon

r = current range between firer and target

Utilizing one of the above formulas to calculate A_{ij}'s, the attrition during each time step was computed using the Euler-Cauchy differencing equations to approximate Lanchester's *force-on-force* attrition differential equations.

4. Termination of Ground Attack

The ground attack is terminated when either:

(1) One of the two opponent forces is annihilated;

(2) A distance between each attacking subunit and each defensive subunit which is still engaging becomes "too close";

(3) Any attacking subunit passes by the flanks of the forward most defensive subunit still in the battle.

The criteria for being "too close" is user input. This allows for flexibility of breakpoint distance for various weapon systems on the battlefield.

E. THE PARAMETRIC TERRAIN

The terrain affects a great deal on detection, mobility, tactics, and intervisibility between weapon systems in ground combat environments. In the model, the battle is simulated on 3 x 4 Km piece of terrain represented as a part of the coastal area east of the Korean Peninsula. It is important to have a terrain representation to emulate actual terrain areas. The model uses the parametric terrain representation method which was proposed by Chris Needle [Ref. 8]. The idea of the parameterized terrain is that the elevation of any hill mass can be represented by a bivariate normal density function. Mathematically, if $f_I(X,Y)$ is a function giving the elevation of the I's hill masses at any X,Y map coordinates on the battlefield, the overall terrain elevation at X,Y is obtained as the positive maximum over all the hill masses,

$$Z = f(X,Y) = \underset{I = 1, \ldots, NHILLs}{\text{maximum}} f_I(X,Y)$$

where NHILLs is the total number of hill masses on the battlefield. Then, elevation data is used to compute the existence of line of sight between opposing forces which is a key element in detection process. The model uses the line-of-sight routine which was written by Prof. James K. Hartmann [Ref. 5].

In order to represent a piece of real terrain with parametric terrain, it is necessary to fit hill mass functions $f_I(X,Y)$ to a contour map of the terrain to be modeled. The fitting

process can be done by comparing a computer generated contour map by varying the bivariate normal parameters to the original terrain map. The computer generated terrain map of the battle area is inclosed as Figure 6. Appendix D presents the program listing for plotting a contour map from hill mass functions. This program can be used for the user to fit a specific real terrain which he has in mind into the parameterized terrain.

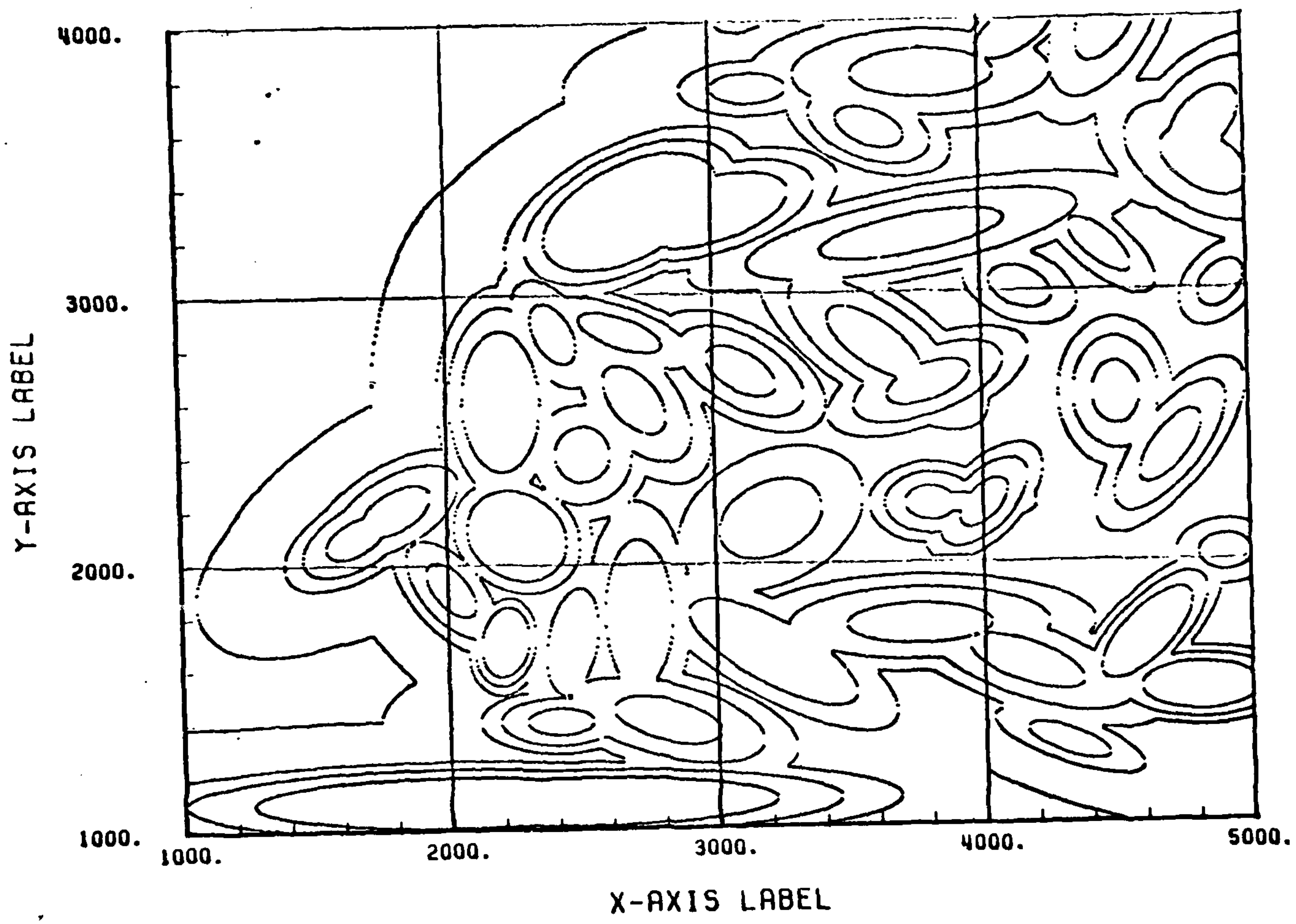


Figure 6. Model Terrain Map

IV. FINAL REMARKS

In order to illustrate basic modeling points for understanding and developing complex-operational amphibious-warfare models, a simplified model (specially tailored to a small-unit-amphibious-operation scenario) has been developed. Although several user options and varieties of modeling techniques have been incorporated into the model, the model should not be viewed as the final product.

There are several areas of model enhancement and enrichment that should be considered in the future. Only the aggregated amphibious task force fire support is played explicitly in the amphibious assault phase. The effect of artillery, naval gunfire and close air support during the various phases of amphibious operations should be added to the model. Ammunition consumption and resupply are considered vital to the success of any military operation, and such aspects should be also added to the model. Enemy forces should be played in greater detail. The movement of reacting enemy force and their dynamic-defensive-position selection are being considered for future model inclusion. A module which considers enemy reaction time, the terrain, and the tactical situation, and which dynamically determines which side is going to take the defensive role, as well as its defensive position, has been proposed to enhance the existing model. Inclusion of these features would create a more complicated model, adding realism but detracting from the current simple and transparent form.

The model currently simulates the ground combat on a 3 x 4 Km piece of coastal terrain representing an area of Ham-Hung, Korea. In order to simulate combat on a user's specific terrain, he needs to fit a parameterized terrain model to the actual terrain of interest. Doing so using the terrain fitting method described in the preceding chapter is an extremely tediuous and time consuming process. The development of a more efficient terrain fitting technique will greatly enhance the model flexibility, and increase its utility in responding to interrogations from the real world.

The interested reader may obtain the model program deck and the sample input data deck from Prof. James G. Taylor, Naval Postgraduate School. Since the data used for the model run is hypothetical and greatly simplified, the reader is cautioned not to draw any analytical conclusions from the output.

APPENDIX A

SAMPLE EXPECTED OUTPUT

I. Amphibious Assault Summary

AMPHIBIOUS ASSAULT INFORMATION

INITIAL FORCE STRENGTH

WAVE	1	2	3	4	5
LVA	10.0	11.0	11.0	10.0	3.0

DT = 3.0 DS = 1.0

LVA ENGR SPECS

SPDMAX	SPDMIN	HTMAX	HTMIN	WID
10.30	3.50	1.70	0.60	3.53

DEFENSIVE TACTICAL PARAMETERS

	RANGE		AIM-RELOAD	PROJECTILE
	MAX	MIN	TIME	VELOCITY
TANK	1500.0		15.00	350.00
ATGM	2000.0	200.0	30.00	350.00

DEFENSIVE TACTICAL ALLOCATION WEIGHTS:
WAVE 1 = 2.00 WAVE 2 = 1.00

DEFENSIVE FORCE ATTRITION COEFFICIENTS

	ALPHA*A	BETA*A
DT	0.00006	0.00070
DS	0.00008	0.00090

WBETA(1) = 0.00050 WBETA(2) = 0.00070

BREAKPOINT ASSUMPTION: 0.3*(TOTAL DEP FORCE)

DEFENSIVE FORCE LEVEL FOR GROUND ATK 0.32

DISPERSION DATA

RANGE	TSIGV	RANGE	TSIGH	RANGE	TMEANH
25.0	0.0	25.0	0.0	25.0	0.0
500.0	2.0	500.0	2.0	500.0	1.0
1000.0	5.0	1000.0	5.0	1000.0	5.0
2000.0	20.0	2000.0	20.0	2000.0	10.0
5000.0	25.0	5000.0	25.0	5000.0	15.0
10000.0	25.0	10000.0	25.0	10000.0	15.0

RANGE	SSIGV	RANGE	SSIGH
25.0	0.0	25.0	0.0
250.0	5.0	250.0	5.0
500.0	7.5	500.0	7.5
1000.0	14.0	1000.0	14.0
2500.0	15.5	2500.0	15.5
5000.0	17.0	5000.0	17.0
10000.0	20.0	10000.0	20.0

II. Assault Phase Time Step Summary

TIME = 815.0 SECONDS

WAVE	FORCE LEVEL	STATUS	LOST-PCT	TSURV
1	0.7322	1	0.927	
2	1.7451	1	0.841	
3	4.5060	1	0.590	
4	9.8051	3	0.019	
5	3.0000	2	0.0	
TANK	0.0		1.000	19.79
ATGM	0.0		1.000	0.0

FINAL LVA SURVIVORS ASHORE = 19.788
GROUND ATTACK STARTS AFTER DEFENDER BROKE CONTACT

GROUND ATK TIME = 825.0

III. Initial Ground Combat Summary

INITIAL GROUND COMBAT INFORMATION

UNIT	X	Y	FORCE LEVEL
1	2000.0	1900.0	3.0
2	1900.0	2400.0	3.0
3	1500.0	2100.0	3.0
4	3800.0	2700.0	3.0
5	3800.0	2300.0	3.0
6	3600.0	1700.0	3.0

ATTRITION IS DETERMINISTIC

ROUTES DETERMINED BY USER

ATTACK VEHICLE SPEED IS 12.0

BREAKPOINT DISTANCE IS 500.0

DEFENDER WILL MOVE TO ALTERNATE POSITIONS
ALTERNATE POSITIONS ARE:

UNIT	X	Y
4	4500.0	3800.0
5	4500.0	2700.0
6	4600.0	1800.0

ATK KILL PROBABILITIES

RANGE	P	PHH	PHM	PKH
500	0.85	0.85	0.75	0.70
1000	0.80	0.80	0.75	0.70
1500	0.75	0.75	0.70	0.65
2000	0.60	0.65	0.60	0.55
2500	0.45	0.50	0.50	0.35
3000	0.20	0.20	0.20	0.20

DEF. KILL PROBABILITIES

RANGE	P	PHH	PHM	PKH
500	0.60	0.70	0.65	0.85
1000	0.85	0.90	0.85	0.90
1500	0.80	0.85	0.85	0.80
2000	0.75	0.80	0.75	0.70
2500	0.60	0.70	0.65	0.65
3000	0.40	0.45	0.40	0.50

IV. Ground Combat Time Step Summary

TIME = 1395 SECONDS

UNIT	X	Y	FORCE LEVEL	STATUS	LOST-PCT	TARGETS
1	2420.8	1984.2	0.0	2	1.000	
2	3664.1	2253.0	0.0	2	1.000	
3	4397.4	1742.2	2.9	0	0.038	
4	4500.0	3800.0	0.0	2	1.000	
5	4500.0	2700.0	2.8	0	0.055	
6	4600.0	1800.0	0.0	2	1.000	

* DISTANCE BETWEEN FORCES IS TOO CLOSE. END OF BATTLE

APPENDIX B

LISTING OF SAMPLE INPUTS

I. Amphibious Assault Input

```
  1    1
  10.30       3.5        1.7        0.6        3.533
  10.
1500.       2000.       200.
  15.         30.       350.       350.
  25.        500.      1000.      2000.       5000.      10000.       0.
   2.          5.        20.        25.         25.
  25.        500.      1000.      2000.       5000.      10000.       0.
   2.          5.        20.        25.         25.
  25.        500.      1000.      2000.       5000.      10000.       0.
   1.          5.        10.        15.         15.
  25.        250.       500.      1000.       2500.       5000. 10000.
   0.          5.         7.5       14.         15.5        17.     20.
  25.        250.       500.      1000.       2500.       5000. 10000.
   0.          5.         7.5       14.         15.5        17.     20.
   2.          1.
 10.         11.        11.        10.          3.
   3.          1.
  0.00006    0.00008
  0.0007     0.0009
  0.0005     0.0007
  .32
   50.        100.
```

II. Terrain Data

```
  46
  0.
2000.       1100.       170.        0.1       999.9       8.0
1800.       2200.       150.       30.        350.        2.0
2000.       1900.       150.      130.        300.        2.
2400.       1400.       150.        0.1       300.        2.5
2450.       1700.       130.       80.        500.        2.2
2700.       1800.       138.       90.        500.        2.2
3200.       1650.       140.      150.        600.        3.
4300.       1300.       130.      160.        400.        3.5
3750.       1750.       150.        0.1       660.        3.6
4150.       1600.       150.      160.        550.        3.
3200.       2150.       130.       25.        500.        1.5
4600.       1700.       170.       45.        300.        2.5
4800.       1500.       170.        0.1       300.        2.5
2200.       2600.       170.       90.        350.        1.8
2400.       2850.       150.      120.        300.        1.8
3100.       2700.       150.      150.        350.        2.
```

```
2500.      2400.      150.       0.1     250.     1.0
2650.      2850.      150.     160.      400.     3.0
2700.      2600.      150.     130.      370.     1.8
3800.      2200.      150.       0.1     230.     1.5
4500.      2600.      150.      90.      280.     1.3
3600.      2800.      150.     145.      500.     2.5
2700.      3300.      190.      25.      350.     2.0
3000.      3300.      170.      15.      400.     2.5
3150.      3750.      130.       0.1     350.     2.5
3750.      3200.      150.      10.      850.     5.0
3800.      3800.      150.       0.1     650.     3.
3600.      3600.      150.     160.      320.     3.0
4150.      3950.      170.      30.      220.     2.2
1650.      2100.      150.      30.      300.     2.0
2250.      2100.      180.     150.      220.     1.2
4000.      2200.      150.      45.      280.     2.
3900.      2200.      150.       0.1     300.     3.5
   0     0     0     0     0     0     1     7    18    27
   0    33    39    53    62     0    74    77    83    93
   0     0     0     0     0     0     6    11     9     6
   0     6    14     9    12     0     3     6    10     9
 101
   1     2     3    30     4    43     1     3     4     5
   6    32    33     7    11    31    43     1     6     7
   8     9    10    11    33    43    10    12    13     9
   8    42     2    14    30    23    15     3    14    15
  16    17    18    19    20     3     6    23    11     7
   2    31    11    16    20    22    34    35    44    45
  46    20    21    22    12    34    35    36    40    41
  42    45    46    14    23    15    23    24    25    15
  26    14    25    26    27    28    29    24    22    23
  35    44    26    27    29    35    36    37    38    39
  40
```

III. Ground Combat Input

```
1 28943
03 03
0000.0 2500.0 0500. 4000.0
3.0 3.0 3.0
1 2
2000.0 1900.0
1900.0 2400.0
1500.0 2100.0
01
5000.0 2500.0
01
4900.0 2150.0
02
2200.0 1700.0
4800.0 1750.0
3800.0 2700.0 3.0 190 120
```

```
3800.0 2300.0 3.0 190 120
3600.0 1700.0 3.0 180 120
0 0500.0  4
4500.0 3800.0
4500.0 2700.0
4600.0 1800.0
0.85 0.85 0.75 0.70
0.80 0.75 0.70 0.65
0.75 0.75 0.70 0.65
0.60 0.65 0.60 0.55
0.45 0.50 0.50 0.35
0.20 0.20 0.20 0.20
0.60 0.70 0.65 0.85
0.85 0.90 0.84 0.90
0.80 0.85 0.85 0.80
0.75 0.80 0.75 0.70
0.60 0.70 0.65 0.65
0.40 0.45 0.40 0.50
```

APPENDIX C

DEFINITION OF VARIABLES IN COMPUTER PROGRAM

1. The Amphibious Assault Phase

CDSURV(I) = Current strength of defensive force I

I = 1 TANK
I = 2 ATGM

CSURV(I) = Current strength of assault wave I

DA(I) = Attrition rate for def. unit I due to the effects of ATFFS/TLF

DS1 = That portion of the DS unit assigned to engaging the closer of two multiple waves in the ATGM engagement window

DS2 = That portion of the Ds unit assigned to engaging the farther of two multiple waves in the ATGM engagement window

DT1 = That portion of the Dt unit assigned to engaging the closer of two multiple waves in the TANK engagement window

DT2 = That portion of the DT unit assigned to engaging the farther of two multiple waves in the TANK engagement window

DT1PH = Hit probability of rounds fired by DT1 against wave TENG(1)

DT1ROF = Rate of fire utilized by DT1 against wave TENG(1)

DINIT = Initial strength of def. force I

IL(I) = When equal to 1 indicates the wave landed shore

IWPN = Weapon code: TANK = 1, ATGM = 2

IWSTAT(I) = Current status of wave I
0 - not engaging
1 - landed
2 - under fire by ATGM
3 - under fire by TANK
4 - under fire both ATGM and TANK

GALF	=	Denote whether the LF build-up is sufficient for the ground attack 0 - not sufficient 1 - sufficient
GATK	=	Denote whether the LF initiated the ground attack 0 - not started yet 1 - started already
GATM	=	Time when the ground attack started
RD	=	Distance offshore at which waves initiate transition
RKSURV(I)	=	Concatenation of CSURV and CDSURV
SA(I)	=	Attrition rate for wave I due to ATGM
SENG(I)	=	The wave number of the closer of two waves in the ATGM engagement window
SRNG(I)	=	Firing range to wave SENG(I)
SSIGH	=	The std dev error in the horizontal for ATGM
SSIGV	=	The std dev error in the vertical for ATGM
SWTS(I)	=	The proportion of the total DS strength to be allowed to engaging SENG(I)
TA(I)	=	Attrition rate for wave I due to TANK
TBW	=	The interarrival time between waves arriving at the beach
TMEANH	=	The bias error in the horizontal for TANK
TMEANV	=	The bias error in the vertical for TANK
TENG(I)	=	The wave number of the closer of two waves in the tank engagement window
TRNG(I)	=	the firing range to wave TENG(I)
TSIGH	=	The std dev error in the horizontal for TANK
TSIGV	=	The std dev error in the vertical for TANK
TSURV	=	Total number of surviving LVA at the current time
TWTS(I)	=	The proportion of the total DT strength to be allowed to engaging TENG(I)
WVINT(I)	=	Initial strength of wave I

2. The Ground Attack Phase

ALPHA(I)	=	Initial attrition-rate coefficient for stochastic attrition module
APOA(I,J)	=	The average proportion of the j^{th} attacker of unit i allocated to fire on unit i
AVSP	=	Average speed of moving attacking units
BREAK	=	Breakpoint distance between attacking units and defenders
DISMAX	=	Maximum distance allowed between attacking units before the leading unit is delayed
DIST	=	The straight-line distance between two movement nodes inputed by the user
DST	=	The distance in meters to be moved each time step by attacking units
FL(I)	=	Force level of unit i
FO(I)	=	Force level of unit i
FO(I)	=	Initial force level of unit i
IALT	=	Denotes whether the user desires alternate defensive positions or not 0 - yes 1 - no
IC	=	Counts number of time units a defender has been moving
IDIR(I,J)	=	Direction of j^{th} interval in i^{th} route
II(I)	=	Interval index for unit i
IMOVE	=	Number of time units a defender is allowed for moving to an alternate position
IPRDIR(I)	=	Primary direction of movement for unit i
IRTE	=	Denotes whether user wants to input routes or not 0 - program determined routes 1 - user determined routes
IS	=	Random number seed used for stochastic attrition
ISECWD(I)	=	Width of search sector for unit i

ISPD = Input variable to denote user's desired speed for attackers movement

1 - 9 mph
2 - 12 mph
3 - 15 mph
4 - 18 mph

ITEM = Input variable denoting number of time steps allowed for defender's move

ITIME = Current time, in seconds, of battle

ITRIT = Input variable denoting whether attrition will be stochastic or deterministic

0 - stochastic
1 - deterministic

IUSTAT(I) = Current status of unit it

0 - unit alive, not firing
1 - unit alive and firing
2 - unit killed
3 - unit moving

LOA(I,J) = The number of the j^{th} attacker of unit i

LOST(I,J) = Denotes whether line-of-sight exists between unit i and j or not

LOT(I,J) = The number of the j^{th} target of unit i

MVTDIR(I) = Movement direction of unit i

N(I) = Number of nodes inputed by user for route i

NA(I) = Number of attackers of unit i

NBU = Number of defense units

NF(I) = Number of time units i is allowed to fire at the same location

NLOSC(I,J) = Number of continuous time steps that line-of-sight does not exist between unit i and unit j

NOI(I) = Number of intervals in the i^{th} route

NRU = Number of attack units

NT(I) = Number of targets of unit i

OFL(I) = Force level of unit i during previous time step

P(I,J)	=	Probability of 1^{st} round hit by unit i in range band j
PHH(I,J)	=	Probability of a hit following a hit by unit i in range band j
PHM(I,J)	=	Probability of a hit following a miss by unit i in range band j
PKH(I,J)	=	Probability of a kill given a hit by unit i in range band j
PM	=	The proportion of time a moving unit is searching for targets
POA(I,J)	=	The proportion of the j^{th} attacker of unit i allocated to fire on unit i
POL(I)	=	Percent of unit i lost during the current time step
PTT(I)	=	Proportion of surviving firepower allocated to the i^{th} target if there are j targets available
RANGE	=	Current minimum distance between attackers and defenders
Q(I,J)	=	Probability that unit j is not detected by unit i at current time
RF	=	Detection rate reduction factor for a firing unit (in comparison with non-firing unit)
RMINTK	=	Minimum effective range for defending weapon system
RMINTW	=	Minimum effective range for attacking weapon system
RMXTK	=	Maximum effective range for defending weapon system
RMXTW	=	Maximum effective range for defending weapon system
ROT(I,J)	=	The range of the j^{th} target of unit i
SIZETK	=	Size of attacking vehicle
SIZETW	=	Size of defending vehicle
TA(K)	=	Time to acquire a target for k^{th} weapon system type (k = 1,2)
TF1(K)	=	Time of flight to 1000m for k^{th} weapon systen type (k = 1,2)

TF2(K)	=	Time of flight to 2000m for k^{th} weapon system type (k = 1,2)
TF3(K)	=	Time of flight to 3000m for k^{th} weapon system type (k = 1,2)
TH(K)	=	Time to fire a round following a hit for weapon system type k (k = 1,2)
TI(K)	=	Time to fire first round after target has been acquired for weapon system type k (k = 1,2)
TM(K)	=	Time to fire a round following a miss for weapon system type k (k = 1,2)
TNKFR	=	Firing rate for attacking weapon system
TOWFR	=	Firing rate for defending weapon system
TPOL(I)	=	Total percentage of lost since battle began for unit i
VISFR(I,J)	=	The fraction of unit i seen by unit j
VISFRA	=	Fraction of unit A as seen by unit B
VISFRB	=	Fraction of unit B as seen by unit A
X(I),Y(I)	=	Coordinates of unit i
XA(I),YA(I)	=	Coordinates of alternate position for defender i
XIC(I,J) YIC(I,J)	=	Coordinates of the j^{th} interval endpoint of the route for unit i
XL,YL	=	Distance added to previous interval endpoint for vehicle to move DST during a time step
XLOC(I,J) YLOC(I,J)	=	Coordinates of the j^{th} node inputed by the user for the route of unit i

APPENDIX D

PROGRAM LISTING

```
C AMPHIBIOUS ASSAULT PHASE
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      CCMMON /DISPER/TSIGV(6,2),TSIGH(6,2),TMEANH(6,2),
     1SSIGV(7,2),SSIGH(7,2)
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,
     1SVEL,DEFWTS(2)
      COMMON /SUPEFT/GAMMA,DELTA
      CCMMON /IOUT/ISURV,IATTR
C GRCUND ATTACK PHASE
      COMMON /GRP1/ IPRDIR(6),ISECWD(6),MVTDIR(6),X(6),Y(6),SPD(6)
      CCMMON /GRP2/ TA(2),T1(2),TH(2),TM(2),TF1(2),TF2(2),TF3(2),
     1P(2,6),PHH(2,6),PHM(2,6),PKH(2,6),TF(2)
      COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),
     1IDIR(3,200),AVSP,ISPD
     1,IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZETK,
     1SIZETW,NT(6),NF(6),SRF,DISMAX,
     1NLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,
     1PTT(3,3),RF,POA(6,6),APOA(6,6),LOA(6,6),NA(6),OFL(6),POL(6)
      COMMON /GRP4/ TPOL(6),OLDQ(6,6),Q(6,6)
      COMMON /GRP5/ LOT(6,6),ROT(6,6)
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100),SPRD(100)
      COMMON /HILLS/ ECC(100),PXX(100),PYY(100),PXY(100),BASE
      COMMON /HILLS/ NHILLS
      COMMON /COVER/ CXC(150),CYC(150),CPEAK(150),CPXX(150),CPYY(150)
      CCMMON /COVER/ CPXY(150),NCVELS
      COMMON /COUNTR/ KH,KHW,KV,KN,KGRS,KELL,KINT
      CCMMON /GRID/ LST(5,4),NHL(5,4),LISTH(150),KHREP(150),KTREP
      COMMON /GRID/ LSTC(5,4),NC(5,4),LISTC(400),KCREP(150)
      COMMON /GRP6/ ALPHA(6)
      COMMON /GRP7/ XA(6),YA(6),IMOVE(6)
C
      GATM=0.
      GATK=0.
      CALL DATAIN
      CALL SETUP
      CALL SEA(GATM,GATK)
      IF(GATK.NE.0.) GO TO 106
      WRITE(6,105)
  105 FORMAT(1X,'TOTAL LANDED LF STRENGTH IS NOT SUFFICIENT FOR GATK')
```

```
      STOP
  106 IF(GATK-2.0) 10,20,30
   10 WRITE(6,107)
  107 FORMAT(1X,'GROUND ATTACK STARTS WHILE SHORE COMBAT IS GOING ON')
      GO TO 110
   20 WRITE(6,108)
  108 FORMAT(1X,'GROUND ATTACK STARTS AFTER DEFENDER BROKE CONTACT')
      GO TO 110
   30 WRITE(6,109)
  109 FORMAT(1X,'GROUND ATTACK STARTS AFTER ALL WAVES LANDED')
  110 WRITE(6,111) GATM
  111 FCRMAT(/1X,'GROUND ATK TIME=',F6.1)
      CALL GROUND(GATM)
      STOP
      END
C
      SUBROUTINE SEA(GATM,GATK)
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      CALL OUTPUT
      IRD=500
      ITBW=120
      RD=1.0*IRD
      TBW=1.0*ITBW
      TINT=0.0
C
C  COMPUTATION OF FIRST WAVE TIME PARAMETERS
C     TA-TIME FIRST WAVE INITIATES TRANSITON
C     TB-TIME FIRST WAVE COMPLETES TRANSITION
C     TFF-TIME FIRST WAVE REACHES THE BEACH
C
      TAA=(5000.-RD)/SPDMAX
      TB=TAA+TTS
      TFF=TB+(RD-(0.5*(SPDMAX-SPDMIN)*TTS)-150.)/SPDMIN
      DEL=10.
C     WRITE(6,55) RD,TBW
C  55 FORMAT(/,1X,'ITERATION INITIATED...RD=',F10.3,1X,'TBW=
C    1',F10.3)
      CALL RKINT(DEL,TINT,N,GATM,GATK)
      RETURN
      END
      SUBROUTINE RKINT(H,TI,N,GATM,GATK)
C
C  SUBROUTINE RKINT PROVIDES THE INTERFACE BETWEEN
C  THE EULER NUMERICAL INTEGRATION ROUTINE(RKLDEQ)
C  AND THE SUBROUTINE ATTR WHICH DETERMINES EACH
C  UNIT'S STATUS AS TIME PROGRESSES THROUTH THE
```

```
C   AMPHIBIOUS OPERATION                                                    GRA00900
C                                                                           GRA00910
         CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,       GRA00920
        1TBW,DINIT(2),GAINL,IWSTAT(5)                                       GRA00930
         CCMMON /IOUT/ISURV,IATTR                                           GRA00940
         DIMENSION CSURV(5),CDSURV(2),TA(5),SA(5),DA(2)                     GRA00950
         DIMENSION  RKSURV(7),RKATTR(7),TATTR(200,12),TIME(200)             GRA00960
C                                                                           GRA00970
C   IMAX - MAXIMUM ALLOWABEL NUMBER OF TIME INTERVALS                       GRA00980
C   ITE - A SWITCH VARIABLES SET TO 1 WHEN THE DEF.TANK                     GRA00990
C         UNIT INITIATES ITS FIRE                                           GRA01000
C   ISE - A SWITCH VARIABLES SET TO 1 WHEN THE DEF.ATGM                     GRA01010
C         UNIT INITIATES ITS FIRE                                           GRA01020
C   T - CURRENT TIME                                                        GRA01030
C   IT - CURRENT TIME PERIOD                                                GRA01040
C                                                                           GRA01050
         GALF=0.                                                            GRA01060
         IMAX=199                                                           GRA01070
         ITE=0                                                              GRA01080
         ISE=0                                                              GRA01090
         TSURV=0.                                                           GRA01100
         TIME(I)=0.                                                         GRA01110
         T=TI                                                               GRA01120
         DO 10 I=1,5                                                        GRA01130
         CSURV(I)=WVINT(I)                                                  GRA01140
         TSURV=TSURV+CSURV(I)                                               GRA01150
         IL(I)=0                                                            GRA01160
         IWSTAT(I)=0                                                        GRA01170
      10 CONTINUE                                                           GRA01180
         DO 15 I=1,2                                                        GRA01190
         CDSURV(I)=DINIT(I)                                                 GRA01200
      15 CONTINUE                                                           GRA01210
         DO 20 J=1,12                                                       GRA01220
      20 TATTR(1,J)=0.                                                      GRA01230
         IT=1                                                               GRA01240
         DO 25 I=1,5                                                        GRA01250
      25 RKSURV(I)=CSURV(I)                                                 GRA01260
         RKSURV(6)=CSURV(1)                                                 GRA01270
         RKSURV(7)=CSURV(2)                                                 GRA01280
         DO 30 I=1,7                                                        GRA01290
      30 RKATTR(I)=0.                                                       GRA01300
         NT=0                                                               GRA01310
    1000 CALL ATTR(T,CSURV,CDSURV,TA,SA,DA,GALF,GATK,GATM)                  GRA01320
         IF(IL(1).EQ.99) GO TO 1200                                         GRA01330
         DC 40 I=1,5                                                        GRA01340
         RKSURV(I)=CSURV(I)                                                 GRA01350
      40 RKATTR(I)=(TA(I)+SA(I))*(-1.0)                                     GRA01360
         DO 45 I=1,2                                                        GRA01370
```

```
      RKSURV(I+5)=CDSURV(I)
   45 RKATTR(I+5)=-1.0*DA(I)
      S=RKLDEQ(7,RKSURV,RKATTR,T,H,NT)
      DC 50 I=1,5
      CSURV(I)=RKSURV(I)
   50 CONTINUE
      DO 55 I=1,2
      CDSURV(I)=RKSURV(I+5)
   55 CONTINUE
      IF(S-1.) 1100,1000,1200
 1100 WRITE(6,60)
   60 FCRMAT(1X,'ERRO..S.NE.1.OR.2')
      STOP
 1200 CONTINUE
      IT=IT+1
      TSURV=0.
      DO 65 L=1,5
  65  TSURV=TSURV+CSURV(L)
      IF(TSURV.LE.0.) TSURV=0.
      TIME(IT)=T
C  PRINT RESULT OF SHIP TO SHORE MOVEMNET AFTER EACH TIME STEP
      WRITE(6,112) T
  112 FORMAT(///1X,'TIME=',F6.1,1X,'SECONDS'//)
      WRITE(6,113)
  113 FORMAT(1X,'WAVE',2X,'FORCE LEVEL',2X,'STATUS',2X,'LOST-PCT',
     12X,'TSURV')
      DO 66 I=1,4
      PLOST=1.-CSURV(I)/WVINT(I)
      WRITE(6,114) I,CSURV(I),IWSTAT(I),PLOST
  114 FORMAT(3X,I1,3X,F10.4,5X,I1,5X,F8.3)
   66 CONTINUE
      PLOST=1.-CSURV(5)/WVINT(5)
      WRITE(6,115) CSURV(5),IWSTAT(5),PLOST,TSURV
  115 FORMAT(3X,'5',3X,F10.4,5X,I1,5X,F8.3,2X,F5.2)
      PLOST=1.-CDSURV(1)/DINIT(1)
      WRITE(6,116) CDSURV(1),PLOST
  116 FORMAT(1X,'TANK',2X,F10.4,11X,F8.3)
      PLOST=1.-CDSURV(2)/DINIT(2)
      TASURV=CDSURV(1)+CDSURV(2)
      WRITE(6,117) CDSURV(2),PLOST,TASURV
  117 FORMAT(1X,'ATGM',2X,F10.4,11X,F8.3,2X,F5.2)
C
      DO 80 J=1,5
      TATTR(IT,J)=TA(J)
   80 TATTR(IT,J+5)=SA(J)
      DO 85 J=1,2
   85 TATTR(IT,J+10)=DA(J)
      R=RNG(T-4.*TBW)
```

```
C   DETERMINE IF ALL WAVES LANDED AND GROUND ATK STARTED                 GRA01860
        IF(R.LT.75.) GO TO 2000                                          GRA01870
C                                                                        GRA01880
        IF(IT.GT.IMAX) GO TO 2000                                        GRA01890
        IF(IL(1).EQ.99) GO TO 2000                                       GRA01900
        GO TO 1000                                                       GRA01910
 2000   N=IT                                                             GRA01920
        WRITE(6,90) TSURV                                                GRA01930
   90   FORMAT(1X,'FINAL LVA SURVIVORS ASHORE= ',F10.3)                  GRA01940
        IF(GATK.GE.1.) GO TO 2222                                        GRA01950
        IF(TSURV.LT.9.) GO TO 2222                                       GRA01960
        GATK=3.                                                          GRA01970
        GATM=T                                                           GRA01980
 2222   RETURN                                                           GRA01990
        END                                                              GRA02000
        FUNCTION RKLDEQ(N,Y,F,X,H,NT)                                    GRA02010
        DIMENSION Y(1),F(1),Q(25)                                        GRA02020
        NT=NT+1                                                          GRA02030
        GO TO (1,2,3,4),NT                                               GRA02040
    1   H1=H                                                             GRA02050
        AA=H1/4.0                                                        GRA02060
        DO 11 J=1,N                                                      GRA02070
   11   Q(J)=0.                                                          GRA02080
        X=X+AA                                                           GRA02090
        GO TO 5                                                          GRA02100
    2   X=X+AA                                                           GRA02110
        GO TO 5                                                          GRA02120
    3   X=X+AA                                                           GRA02130
        GC TO 5                                                          GRA02140
    4   DO 93 L=1,N                                                      GRA02150
   93   Y(L)=Y(L)+AA*F(L)                                                GRA02160
        NT=0                                                             GRA02170
        X=X+AA                                                           GRA02180
        RKLDEQ=2.                                                        GRA02190
        GO TO 6                                                          GRA02200
    5   DO 90 I=1,N                                                      GRA02210
   90   Y(I)=Y(I)+AA*F(I)                                                GRA02220
        RKLDEQ=1.0                                                       GRA02230
    6   RETURN                                                           GRA02240
        END                                                              GRA02250
C                                                                        GRA02260
        SUBROUTINE ATTR(T,CSURV,DSURV,TA,SA,DA,GALF,GATK,GATM)           GRA02270
C                                                                        GRA02280
C   GIVEN THE CURRENT TIME AND STATE VARIABLE STRENGTHS,                 GRA02290
C   SUBROUTINE ATTR DETERMINES THE ATTRITION RATES AND UPDATES           GRA02300
C   THE STATUS OF EACH UNIT WITH RESPECT TO SHORE MOVEMENT               GRA02310
C   AND IMPLEMENTS THIS INFORMATION INTO THE ATTRITION LOSS RATE         GRA02320
C   COMPUTATION.                                                         GRA02330
```

```
C   TA(I) - CURRENT ATTRITION LOSS RATE FOR WAVE I DUE TO TANK FIRE
C   SA(I) - CURRENT ATTRITION LOSS RATE FOR WAVE I DUE TO ATGM FIRE
C   DA(I) - CURRENT ATTRITION LOSS RATE FOR DEF. FORCE I DUE TO
C           ATFFS(AMPHIBIOUS TASK FORCE FIRE SUPPORT)/TLF EFFECTS
C
C
      COMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,
     1SVEL,DEFWTS(2)
      INTEGER TENG(2),SENG(2)
      DIMENSION TRNG(2),TWTS(2),SRNG(2),DSURV(2),SWTS(2),
     1CSURV(5),TA(5),SA(5),DA(2)
C
C
      DO 10 I=1,5
      TA(I)=0.
      SA(I)=0.
   10 CONTINUE
C
      DS1=0.
      DS2=0.
      DT1=0.
      DT2=0.
      FAC=1.0
C
C   DETERMINE IF PART OF LANDING FORCE ADVANCE TO ATTACK INLAND
C   KEY TERRAIN
C
      IF(GATK.EQ.1.0) GO TO 2929
      IF(GALF.EQ.1.0.AND.(DSURV(1)+DSURV(2)).LE.GAINL*(DINIT(1)
     1+DINIT(2))) GATM=T
      IF(GALF.EQ.1.0.AND.(DSURV(1)+DSURV(2)).LE.GAINL*(DINIT(1)
     1+DINIT(2))) GATK=1.0
C
C   DETERMINE IF DEF. BREAKPOINT HAS BEEN REACHED
C
 2929 IF((DSURV(1)+DSURV(2)).LT.0.3*(DINIT(1)+DINIT(2)))
     1 GO TO 20
C
C   DETERINE ATTRITION RATE ON DEFENSIVE FORCES BY ATFFS
C   BASED UPON AREA OR AIMED FIRE STATUS
C
      DA(1)=B(1)
      DA(2)=B(2)
      IF(ITE.EQ.0) DA(1)=A(1)*DSURV(1)
      IF(ISE.EQ.0) DA(2)=A(2)*DSURV(2)
      GO TO 30
```

```
   20 DSURV(1)=0.
      DSURV(2)=0.
      DA(1)=0.
      DA(2)=0.
      IF(GATK.EQ.1.) GO TO 3939
      GAT=T
C
C  DETERMINE IF DEF.BREAKPOINT HAS BEEN REACHED BEFORE SUFFICIENT
C  LANDING FORCE IS BUILT UP ON THE SHORE FOR INLAND ATTACK
C
   97 DO 91 I=1,5
      WVRNG=RNG(GAT-TBW*(I-1))
      IF(WVRNG.LT.75.) IL(I)=1
      IF(IL(I).EQ.1) TLF=TLF+CSURV(I)
   91 CONTINUE
      GAT=GAT+10.
      IF(TLF.LT.9.0.AND.IL(5).EQ.1) RETURN
      IF(TLF.LT.9.0.AND.IL(5).NE.1) GO TO 97
      GATK=2.
      GALF=1.
      GATM=GAT
      WRITE(6,220) GATM
  220 FORMAT(/,1X,'GROUND ATK INITIATES AT TIME=',F7.1)
 3939 IL(1)=99
      WRITE(6,25) T
   25 FORMAT(1X,'BEAKPOINT REACHED AT TIME = ',F9.3)
      RETURN
   30 CALL DTGTS(T,TENG,TRNG,TWTS,SENG,SRNG,SWTS,CSURV)
C
C  DETERMINE THE CUMULATIVE NUMBER OF SURVIVING LVA'S
C  THAT HAVE BEEN REACHED THE BEACH - TLF
C
      TLF=0.
      DO 40 J=1,5
      IF(IL(J).EQ.1) TLF=TLF+CSURV(J)
   40 CONTINUE
C
C  DETERMINE IF TLF BUILT UP IS SUFFICENT FOR GROUND ATK
C
      IF(TLF.GE.9.) GALF=1.
      IF(GATK.EQ.1.) TLF=TLF-9.
C
C  ALLOCATE THE FORCE STRENGTH OF TLF BETWEEN THE TWO
C  DEFENSIVE FORCE UNITS
C
      DSUM=DSURV(1)+DSURV(2)
      TLF1=(DSURV(1)/DSUM)*TLF
      TLF2=(DSURV(2)/DSUM)*TLF
```

```
C
C   ADD TO DA1 AND DA2 THE ATTRITION LOSS RATE DUE
C   TO THE EFFECTS OF TLF1 AND TLF2
C
      DA(1)=DA(1)+TLF1*WB(1)
      DA(2)=DA(2)+TLF2*WB(2)
      IF(DSURV(1).LE.0.0) DA(1)=0.
      IF(DSURV(2).LE.0.0) DA(2)=0.0
C
C   DETERMINE IF THERE EXISTS AN INCOMING WAVE IN THE
C   TANK ENGAGEMENT WINDOW I.E. TENG(1).NE.0
C
      IF(TENG(1).EQ.0.) GO TO 100
      ITE=1
C
C   DETERMINE THE TIME SINCE WAVE TENG(1) CROSSED THE
C   5000. METER OFFSHORE MARK -T1
C
      T1=T-TBW*(TENG(1)-1)
      DT1=TWTS(1)*DSURV(1)
      FAC=1.
C
C   DETERMINE THE SUPPRESSION EFFECT TO BE IMPOSED
C   ON THE DT UNIT BASED ON THE ATTRITICN LOSS RATE
C   CURRENTLY IN EFFECT
C
      SUPFAC=DA(1)
C
      CALL RATE(TRNG(1),SPD(T1),1,SUPFAC,DT1ROF)
      CALL PHIT(TRNG(1),WID,HT(T1),1,SUPFAC,DT1PH)
C
C   DETERMINE THE ATTRITION LOSS RATE FOR WAVE TENG(1)
C   DUE TO DT1 FIRES
C
      TA(TENG(1))=DT1PH*DT1ROF*DT1
C
C   DETERMINE IF THERE IS A SECOND INCOMING WAVE THAT
C   IS IN THE TANK ENGAGEMENT WINDOW, IF THERE IS THE
C   ATTRITION RATE COMPUTATIONS ARE SIMILAR IN FORM
C   TC THOSE PREVIOUSLY PERFORMED FOR THE CLOSER WAVE
C
      IF(TENG(2).EQ.0) GO TO 100
      T2=T-TBW*(TENG(2)-1)
      DT2=TWTS(2)*DSURV(1)
      CALL RATE(TRNG(2),SPD(T2),1,SUPFAC,DT2ROF)
      CALL PHIT(TRNG(2),WID,HT(T2),1,SUPFAC,DT2PH)
      TA(TENG(2))=DT2PH*DT2ROF*DT2
C
```

```
C   DETERINE IF THERE EXISTS AN INCOMING WAVE IN THE ATGM
C   ENGAGEMENT WINDOW, IF THERE IS, DETERMINE THE ATTRITION
C   EFFECTS AGAINST THAT WAVE DUE TO ATGM THE ATTRITION
C   RATE COMPUTATION ARE SIMILAR IN FORM TO THOSE FOR THE
C   EFFECTS DUE THE TANK FIRE.
C
  100 IF(SENG(1).EQ.0) GO TO 200
      ISE=1
      S1=T-TBW*(SENG(1)-1)
      DS1=SWTS(1)*DSURV(2)
      SUPFAC=DA(2)
      CALL RATE(SRNG(1),SPD(S1),2,SUPFAC,DS1ROF)
      CALL PHIT(SRNG(1),WID,HT(T1),2,SUPFAC,DS1PH)
      SA(SENG(1))=DS1PH*DS1ROF*DS1
      IF(SENG(2).EQ.0) GO TO 200
      S2=T-TBW*(SENG(2)-1)
      DS2=SWTS(2)*DSURV(2)
      CALL RATE(SRNG(2),SPD(S2),1,SUPFAC,DS2ROF)
      CALL PHIT(SRNG(2),WID,HT(S2),2,SUPFAC,DS2PH)
      SA(SENG(2))=DS2PH*DS2ROF*DS2
  200 RETURN
      END
C
      SUBROUTINE DTGTS(T,TENG,TRNG,TWTS,SENG,SRNG,SWTS,CSURV)
C
C   GIVEN THE CURRENT TIME AND LVA WAVE SURVIVOR POPULATIONS,
C   SUBROUTINE DTGTS DETERMINES THE WAVE NUMBERS THAT ARE
C   TO BE ENGAGED BY THE DT AND DS DEFENSIVE UNITS BASED
C   ON THE ENGEMENT WINDOW CRITERIA
C
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,
     1SVEL,DEFWTS(2)
      INTEGER TENG(2),SENG(2)
      DIMENSION TRNG(2),SRNG(2),TWTS(2),SWTS(2),CSURV(5),DEMO(5)
      DO 10 I=1,2
      TENG(I)=0
      TWTS(I)=0.
      TRNG(I)=0.
      SRNG(I)=0.
      SENG(I)=0
      SWTS(I)=0.
   10 CONTINUE
      JT=0
      JS=0
      TSUM=0.
      SSUM=0.
```

```
      DO 100 I=1,5
      WVRNG=RNG(T-TBW*(I-1))
      IF(WVRNG.LT.75.) IL(I)=1
      IF(WVRNG.LT.75.) IWSTAT(I)=1
C
C  IF THE FIRING RANGE TO A WAVE IS LESS THAN 75 METERS,
C  THE WAVE IS CONSIDERED TO HAVE REACHED A COVERED AND
C  CONCEALED POSITION ON THE BEACH
C
      IF((WVRNG.GT.TENGMX).OR.(CSURV(I).LT.0.05).OR.
     1(WVRNG.LT.75.).OR.(JT.GE.2)) GO TO 50
      JT=JT+1
      TENG(JT)=I
      TWTS(JT)=DEFWTS(JT)*CSURV(I)
      TSUM=TSUM+TWTS(JT)
      TRNG(JT)=WVRNG
   50 IF((WVRNG.GT.SENGMX).OR.(CSURV(I).LT.0.05).CR.
     1(WVRNG.LT.SENGMN).OR.(JS.GE.2)) GO TO 100
      JS=JS+1
      SENG(JS)=I
      SRNG(JS)=WVRNG
      SWTS(JS)=DEFWTS(JS)*CSURV(I)
      SSUM=SSUM+SWTS(JS)
  100 CONTINUE
C
C  DETERMINE WAVE STATUS
C
      DO 20 I=1,2
      DO 25 J=1,5
      IF(IWSTAT(J).NE.1.AND.SENG(I).EQ.J) IWSTAT(J)=2
   25 CONTINUE
   20 CONTINUE
      DO 30 I=1,2
      DO 35 J=1,5
      IF(IWSTAT(J).EQ.1) GO TO 35
      IF(IWSTAT(J).EQ.2.AND.TENG(I).EQ.J) IWSTAT(J)=4
      IF(IWSTAT(J).NE.2.AND.TENG(I).EQ.J) IWSTAT(J)=3
   35 CONTINUE
   30 CONTINUE
C
      IF(TENG(1).EQ.0) GO TO 500
      DO 200 I=1,2
      TWTS(I)=TWTS(I)/TSUM
  200 CONTINUE
  500 IF(SENG(1).EQ.0) RETURN
      DO 600 I=1,2
      SWTS(I)=SWTS(I)/SSUM
  600 CONTINUE
```

```
      RETURN
      END
C
      SUBROUTINE DATAIN
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      COMMON /DISPER/TSIGV(6,2),TSIGH(6,2),TMEANH(6,2),
     1SSIGV(7,2),SSIGH(7,2)
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,
     1SVEL,DEFWTS(2)
      COMMON /SUPEFT/GAMMA,DELTA
      COMMON /IOUT/ISURV,IATTR
      READ(5,50) ISURV,IATTR
      READ(5,100) SPDMAX,SPDMIN,HTMAX,HTMIN,WID
      READ(5,100) TTS
      READ(5,100) TENGMX,SENGMX,SENGMN
      READ(5,100) TARTM,SARTM,TVEL,SVEL
      READ(5,100) ((TSIGV(I,J),I=1,6),J=1,2)
      READ(5,100) ((TSIGH(I,J),I=1,6),J=1,2)
      READ(5,100) ((TMEANH(I,J),I=1,6),J=1,2)
      READ(5,100) ((SSIGV(I,J),I=1,7),J=1,2)
      READ(5,100) ((SSIGH(I,J),I=1,7),J=1,2)
      READ(5,100) (DEFWTS(I),I=1,2)
      READ(5,103) (WVINT(I),I=1,5)
      READ(5,100) (DINIT(I),I=1,2)
      READ(5,101) (A(I),I=1,2)
      READ(5,101) (B(I),I=1,2)
      READ(5,101) (WB(I),I=1,2)
      READ(5,110) GAINL
      READ(5,101) GAMMA,DELTA
  110 FORMAT(F5.2)
   50 FORMAT(2I5)
  100 FORMAT(7F10.3)
  101 FORMAT(2F10.5)
  103 FORMAT(5F10.5)
      RETURN
      END
C
      SUBROUTINE OUTPUT
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /DISPER/TSIGV(6,2),TSIGH(6,2),TMEANH(6,2),
     1SSIGV(7,2),SSIGH(7,2)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,
     1SVEL,DEFWTS(2)
      COMMON /SUPEFT/GAMMA,DELTA
```

```
C
C***   INPUT SUMMARY PRINTOUT
C
       WRITE(6,20)
    20 FORMAT('1',1X,'AMPHIBIOUS ASSAULT INFORMATION')
       WRITE(6,22)
    22 FORMAT(/1X,'INITIAL FORCE STRENGTH')
       WRITE(6,23)
    23 FORMAT(1X,'WAVE',2X,'1',5X,'2',5X,'3',5X,'4',5X,'5')
       WRITE(6,24) (WVINT(I),I=1,5)
    24 FORMAT(1X,'LVA',5(2X,F4.1))
       WRITE(6,21) (DINIT(I),I=1,2)
    21 FORMAT(/1X,'DT =',1X,F3.1,5X,'DS =',1X,F3.1)
       WRITE(6,25)
    25 FORMAT(/1X,'LVA ENGR SPECS')
       WRITE(6,26)
    26 FORMAT(1X,'SPDMAX',2X,'SPDMIN',3X,'HTMAX',2X,'HTMIN',3X,'WID')
       WRITE(6,27) SPDMAX,SPDMIN,HTMAX,HTMIN,WID
    27 FORMAT(2X,F5.2,3X,F5.2,3X,F5.2,3X,F4.2,2X,F4.2)
       WRITE(6,630)
   630 FORMAT(/1X,'DEFENSIVE TACTICAL PARAMETERS')
       WRITE(6,631)
   631 FORMAT(10X,'RANGE',4X,'AIM-RELOAD',3X,'PROJECTILE')
       WRITE(6,632)
   632 FORMAT(8X,'MAX',3X,'MIN',4X,'TIME',7X,'VELOCITY')
       WRITE(6,633) TENGMX,TARTM,TVEL
   633 FORMAT(1X,'TANK',1X,F6.1,9X,F5.2,7X,F6.2)
       WRITE(6,634) SENGMX,SENGMN,SARTM,SVEL
   634 FORMAT(1X,'ATGM',1X,F6.1,1X,F6.1,2X,F5.2,7X,F6.2)
       WRITE(6,50) DEFWTS(1),DEFWTS(2)
    50 FORMAT(/,1X,'DEFENSIVE TACTICAL ALLOCATION WEIGHTS:',
      1/,1X,'WAVE 1 = ',F5.2,1X,'WAVE 2 = ',F5.2)
       WRITE(6,100)
   100 FORMAT(/1X,'DEFENSIVE FORCE ATTRITION COEFFICIENTS')
       WRITE(6,101)
   101 FORMAT(9X,'ALPHA*A',10X,'BETA*A')
       WRITE(6,102) A(1),B(1)
   102 FORMAT(1X,'DT',6X,F7.5,9X,F7.5)
       WRITE(6,103) A(2),B(2)
   103 FORMAT(1X,'DS',6X,F7.5,9X,F7.5)
       WRITE(6,104) WB(1),WB(2)
   104 FORMAT(/1X,'WBETA(1)=',F7.5,1X,'WBETA(2)=',F7.5)
       WRITE(6,105)
   105 FORMAT(/1X,'BREAKPOINT ASSUMPTION: 0.3*(TOTAL DEF FORCE)')
       WRITE(6,770) GAINL
   770 FORMAT(/,1X,'DEFENDER ATTRITION LEVEL ALLOWING GROUN ATTACK',
      1/,1X,F5.2,'*(TOTAL DEFENDER FORCE)')
       WRITE(6,771) GAMMA,DELTA
```

```
  771 FCRMAT(/1X,'ARTM SUP FACTOR=',F5.1,2X,'ERROR SUP FACTOR=',F5.1)       GRA05700
C                                                                            GRA05710
C**** DISPERSION DATA PRINTOUT                                              GRA05720
C                                                                            GRA05730
      IDISP=1                                                                GRA05740
      IF (IDISP.EQ.0) RETURN                                                 GRA05750
      WRITE(6,601)                                                           GRA05760
  601 FORMAT(/1X,'DISPERSION DATA'/)                                        GRA05770
      WRITE(6,602)                                                           GRA05780
  602 FORMAT(3X,'RANGE',2X,'TSIGV',2X,'RANGE',2X,'TSIGH',                   GRA05790
     12X,'RANGE',2X,'TMEANH')                                                GRA05800
      DO 55 I=1,6                                                            GRA05810
      WRITE(6,603) TSIGV(I,1),TSIGV(I,2),TSIGH(I,1),TSIGH(I,2),             GRA05820
     1TMEANH(I,1),TMEANH(I,2)                                                GRA05830
   55 CONTINUE                                                               GRA05840
  603 FORMAT(1X,F7.1,2X,F5.1,1X,F7.1,1X,F5.1,1X,F7.1,1X,F5.1)              GRA05850
      WRITE(6,604)                                                           GRA05860
  604 FORMAT(/3X,'RANGE',2X,'SSIGV',2X,'RANGE',2X,'SSIGH')                 GRA05870
      DO 56 I=1,7                                                            GRA05880
      WRITE(6,605) SSIGV(I,1),SSIGV(I,2),SSIGH(I,1),SSIGH(I,2)             GRA05890
   56 CONTINUE                                                               GRA05900
  605 FORMAT(1X,F7.1,2X,F5.1,1X,F7.1,1X,F5.1)                               GRA05910
      WRITE(6,606)                                                           GRA05920
  606 FORMAT('1','THE AMPHIBIOUS ASSAULT PHASE BEGINS'///)                 GRA05930
      RETURN                                                                 GRA05940
      END                                                                    GRA05950
C                                                                            GRA05960
      SUBROUTINE PHIT(RANGE,W,H,IWPN,SUPFAC,PRHIT)                          GRA05970
      CCMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,          GRA05980
     1TBW,DINIT(2),GAINL,IWSTAT(5)                                           GRA05990
      CCMMON /DISPER/TSIGV(6,2),TSIGH(6,2),TMEANH(6,2),                     GRA06000
     1SSIGV(7,2),SSIGH(7,2)                                                  GRA06010
      CCMMON /SUPEFT/GAMMA,DELTA                                             GRA06020
C                                                                            GRA06030
      PI=ARCOS(-1.0)                                                         GRA06040
      IF(RANGE.LT.25.) STOP                                                  GRA06050
      IF(IWPN.EQ.1) GO TO 50                                                 GRA06060
C   ATGM FIRING DATA COMPUTATIONS                                            GRA06070
      WMEANH=0.0                                                             GRA06080
      WMEANV=0.0                                                             GRA06090
      CALL INTRP(SSIGV,RANGE,WSIGV,7)                                        GRA06100
      CALL INTRP(SSIGV,RANGE,WSIGH,7)                                        GRA06110
C   TANK FIRING DATA COMPUTATIONS                                            GRA06120
   50 WMEANV=0.0                                                             GRA06130
      CALL INTRP(TMEANH,RANGE,WMEANH,6)                                      GRA06140
      CALL INTRP(TSIGV,RANGE,WSIGV,6)                                        GRA06150
      CALL INTRP(TSIGH,RANGE,WSIGH,6)                                        GRA06160
C   CONVERSION TO MILS                                                       GRA06170
```

```
  100 Z=ARSIN(H/RANGE)                                                  GRA06180
      WSIGV=WSIGV*(1.+DELTA*SUPFAC)                                     GRA06190
      WSIGH=WSIGH*(1.+DELTA*SUPFAC)                                     GRA06200
      TGTH=(Z*6400.0)/(2.0*PI)                                          GRA06210
      TGTW=(ARSIN(W/RANGE))*(6400.0/(2.0*PI))                           GRA06220
C                                                                       GRA06230
C  INSTITUTE NORMALITY ASSUMPTIONS TO CCMPUTE HORIZONTAL                GRA06240
C  AND VERTICAL HIT PROBABILITIES                                       GRA06250
C                                                                       GRA06260
      C=-1.0*SQRT(1./2.)                                                GRA06270
      HOR1=((TGTW/2.)-WMEANH)/WSIGH                                     GRA06280
      HOR2=(((-1.0*TGTW)/2.0)-WMEANH)/WSIGH                             GRA06290
      PHITX=1.0                                                         GRA06300
      IF(ABS(HOR1).GT.8.) GO TO 810                                     GRA06310
      PHITX=0.5*(ERFC(C*HOR1)-ERFC(C*HOR2))                             GRA06320
  810 VER1=((TGTH/2.)-WMEANV)/WSIGV                                     GRA06330
      VER2=(((-1.0*TGTH)/2.)-WMEANV)/WSIGV                              GRA06340
      PHITY=1.0                                                         GRA06350
      IF(ABS(VER1).GT.8.) GO TO 820                                     GRA06360
      PHITY=0.5*(ERFC(C*VER1)-ERFC(C*VER2))                             GRA06370
  820 PRHIT=PHITX*PHITY                                                 GRA06380
      RETURN                                                            GRA06390
      END                                                               GRA06400
C                                                                       GRA06410
      SUBROUTINE INTRP(X,ARG,VAL,N)                                     GRA06420
      DIMENSION X(N,2)                                                  GRA06430
C     WRITE(6,777) ARG                                                  GRA06440
C 777 FORMAT(1X,'ARG*****=',F10.3)                                      GRA06450
      IF(ARG.LT.X(1,1)) GO TO 500                                       GRA06460
      DO 50 I=1,N                                                       GRA06470
      IF(ARG.GT.X(I+1,1)) GO TO 50                                      GRA06480
      DIFF=X(I+1,1)-X(I,1)                                              GRA06490
      DELTA=ARG-X(I,1)                                                  GRA06500
      VAL=X(I,2)+(DELTA/DIFF)*(X(I+1,2)-X(I,2))                         GRA06510
      RETURN                                                            GRA06520
   50 CONTINUE                                                          GRA06530
      IF(ARG.GT.X(N,1)) GO TO 600                                       GRA06540
      VAL=X(N,2)                                                        GRA06550
      RETURN                                                            GRA06560
  600 WRITE(6,601)                                                      GRA06570
  601 FORMAT(' ERROR IN INTRP ARG.GT.X(N,2)')                           GRA06580
      STOP                                                              GRA06590
  500 WRITE(6,501)                                                      GRA06600
  501 FORMAT(' ERROR IN INTRP ARG.LT.X(1,1)')                           GRA06610
      STOP                                                              GRA06620
      END                                                               GRA06630
C                                                                       GRA06640
      SUBROUTINE RATE(RANGE,SPEED,IWPN,SUPFAC,ROF)                      GRA06650
```

```
      COMMON /DEF/TENGMX,SENGMX,SENGMN,TARTM,SARTM,TVEL,SVEL
      COMMON /SUPEFT/GAMMA,DELTA
      ROF=0.0
      IF(RANGE.LT.25.) RETURN
      IF(IWPN.EQ.2) GO TO 500
      IF(RANGE.GT.TENGMX) RETURN
      TRTM=TARTM*(1.0+GAMMA*SUPFAC)
      DT=TRTM+RANGE/(TVEL+SPEED)
      ROF=1.0/DT
      RETURN
  500 IF(RANGE.GT.SENGMX) RETURN
      IF(RANGE.LT.SENGMN) RETURN
      SRTM=SARTM*(1.0+GAMMA*SUPFAC)
      DT=SRTM+RANGE/(SVEL+SPEED)
      ROF=1.0/DT
      RETURN
      END
C
C
      FUNCTION SPD(T)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      IF(T.GT.TAA) GO TO 50
      SPD=SPDMAX
      RETURN
   50 IF(T.GT.TB) GO TO 100
      SPD=SPDMIN+((TB-T)/TTS)*(SPDMAX-SPDMIN)
      RETURN
  100 SPD=SPDMIN
      RETURN
      END
C
      FUNCTION HT(T)
      CCMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      IF(T.GT.TAA) GO TO 50
      HT=HTMAX
      RETURN
   50 IF(T.GT.TB) GO TO 100
      HT=HTMIN+((TB-T)/TTS)*(HTMAX-HTMIN)
      RETURN
  100 H1=HTMIN
      RETURN
      END
C
      FUNCTION RNG(T)
      COMMON /AMPH/IL(5),WB(2),A(2),B(2),ITE,ISE,RD,WVINT(5),WID,
     1TBW,DINIT(2),GAINL,IWSTAT(5)
      COMMON /ENGR/ SPDMAX,SPDMIN,HTMAX,HTMIN,TTS,TAA,TB,TFF
      IF(T.GT.TAA) GO TO 50
```

```
      RNG=5000.0-(SPDMAX*T)                                             GRA07140
      RETURN                                                            GRA07150
   50 IF(T.GT.TB) GO TO 100                                             GRA07160
      RNG=RD-0.5*(T-TAA)*(SPDMAX+SPD(T))                                GRA07170
      RETURN                                                            GRA07180
  100 RNG=RD-(((TB-TAA)/2.0)*(SPDMIN+SPDMAX))-((T-TB)*SPDMIN)           GRA07190
      IF(RNG.LT.75.) RNG=0.0                                            GRA07200
      RETURN                                                            GRA07210
      END                                                               GRA07220
C                                                                       GRA07230
C                                                                       GRA07240
C                                                                       GRA07250
      SUBROUTINE GROUND(GATM)                                           GRA07260
      COMMON /GRP1/ IPRDIR(6),ISECWD(6),MVTDIR(6),X(6),Y(6),SPD(6)      GRA07270
      COMMON /GRP2/ TA(2),T1(2),TH(2),TM(2),TF1(2),TF2(2),TF3(2),       GRA07280
     1P(2,6),PHH(2,6),PHM(2,6),PKH(2,6),TF(2)                           GRA07290
      COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),   GRA07300
     1IDIR(3,200),AVSP,ISPD                                             GRA07310
     1,IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZETK,                  GRA07320
     1SIZETW,NT(6),NF(6),SRF,DISMAX,                                    GRA07330
     1NLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,   GRA07340
     1PTT(3,3),RF,POA(6,6),APOA(6,6),LOA(6,6),NA(6),OFL(6),POL(6)       GRA07350
      COMMON /GRP4/ TPOL(6),OLDQ(6,6),Q(6,6)                            GRA07360
      COMMON /GRP5/ LOT(6,6),ROT(6,6)                                   GRA07370
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),SX(100),SY(100),RHO(100) GRA07380
      COMMON /HILLS/ SCALE(100),TWORHO(100),TWOSCL(100),BASE            GRA07390
      COMMON /HILLS/ NHILLS                                             GRA07400
      COMMON /COVER/ CXC(150),CYC(150),CPEAK(150),CPXX(150),CPYY(150)   GRA07410
      COMMON /COVER/ CPXY(150),NCVELS                                   GRA07420
      COMMON /COUNTR/ KH,KHW,KV,KN,KGRS,KELL,KINT                       GRA07430
      COMMON /GRID/ LST(10,10),NHL(10,10),LISTH(450),KHREP(100),KTREP   GRA07440
      COMMON /GRID/ LSTC(10,10),NC(10,10),LISTC(400),KCREP(150)         GRA07450
      COMMON /GRP6/ ALPHA(6)                                            GRA07460
      COMMON /GRP7/ XA(6),YA(6),IMOVE(6)                                GRA07470
C     INITIALIZATION.                                                   GRA07480
C                                                                       GRA07490
      BL=0.0                                                            GRA07500
      RL=0.0                                                            GRA07510
      MP=0                                                              GRA07520
      PAI=3.14159                                                       GRA07530
      ZL=.00001                                                         GRA07540
C                                                                       GRA07550
C  READ TERRAIN DATA FOR LINE OF SIGHT                                  GRA07560
C  CHECK FOR STOCHASTIC OR DETERMINISTIC ATTRITION                      GRA07570
C  ITRIT-ATTRITION MOD 1=DETERMINISTIC                                  GRA07580
C                      0=STOCHASITIC                                    GRA07590
C  IS-SEED NUMBER                                                       GRA07600
C                                                                       GRA07610
```

```
      READ(9,130) ITRIT,IS
  130 FORMAT(I1,1X,I5)
      DO 132 I=1,6
      CALL LRND (IS,YRAN,1,1,0)
      ALPHA(I)=(-2.*YRAN**2)+(2.*YRAN+.3)
C     WRITE(6,799) YRAN,ALPHA(I)
  132 CONTINUE
C 799 FORMAT(2X,'YRAN,ALPHA',F10.5,2X,F10.5)
C
C  READ IN NUMBER OF ATTACK AND DEFENSE UNITS
C
      READ(9,200) NBU,NRU
  200 FORMAT(I2,1X,I2)
C
C  INITIALIZE WEAPON SIZES
C
      SIZETK=2.5
      SIZETW=2.5
C
C   READ IN EFFECTIVE WEAPON RANGES
C
      READ(9,102) RMINTK,RMXTK,RMINTW,RMXTW
  102 FORMAT(F6.1,1X,F6.1,1X,F6.1,1X,F6.1,1X)
C
C  INITIALIZE PM,RF,TOWFR,TNKFR AND NOD
C
      PM=.352
      RF=.5
      TOWFR=.03
      TNKFR=.1
      NOD=2
      DO 101 I=1,NRU
      NOI(I)=125
  101 CONTINUE
      K=NRU+1
      L=NRU+NBU
      DO 111 I=1,L
      II(I)=0
  111 CONTINUE
C
C   READ IN FORCE LEVELS OF EACH ATTACK UNIT
C
      READ(9,103) (FL(I),I=1,NRU)
  103 FORMAT( 3(F3.1,1X))
C
C   CHECK FOR TYPE OF ROUTE DETERMINITION
C
      READ(9,106) IRTE,ISPD
```

```
  106 FORMAT(I1,1X,I1)                                                  GRA08100
      IF(ISPD.EQ.1) AVSP=9.0                                            GRA08110
      IF(ISPD.EQ.1) DST=40.232                                          GRA08120
      IF(ISPD.EQ.2) AVSP=12.0                                           GRA08130
      IF(ISPD.EQ.2) DST=53.643                                          GRA08140
      IF(ISPD.EQ.3) AVSP=15.0                                           GRA08150
      IF(ISPD.EQ.3) DST=67.053                                          GRA08160
      IF(ISPD.EQ.4) AVSP=18.0                                           GRA08170
      IF(ISPD.EQ.4) DST=80.463                                          GRA08180
C                                                                       GRA08190
C   READ IN INITIAL ATTACK UNIT'S LOCATIONS                             GRA08200
C                                                                       GRA08210
      DC 6 I=1,NRU                                                      GRA08220
      READ(9,107) XIC(I,1),YIC(I,1)                                     GRA08230
  107 FORMAT(F6.1,1X,F6.1)                                              GRA08240
    6 CONTINUE                                                          GRA08250
      IF(IRTE.EQ.1) GO TO 108                                           GRA08260
      DO 2 I=1,NRU                                                      GRA08270
      DO 2 J=2,125                                                      GRA08280
      YIC(I,J)=YIC(I,J-1)+DST*(J-1)                                     GRA08290
      XIC(I,J)=XIC(I,J-1)+DST*(J-1)                                     GRA08300
      IDIR(I,J)=0                                                       GRA08310
    2 CONTINUE                                                          GRA08320
      GO TO 109                                                         GRA08330
  108 CALL ROUTE                                                        GRA08340
  109 SUMRO=0.0                                                         GRA08350
      DO 3 I=1,NRU                                                      GRA08360
      FO(I)=FL(I)                                                       GRA08370
      SUMRO=SUMRO+FO(I)                                                 GRA08380
      X(I)=XIC(I,1)                                                     GRA08390
      Y(I)=YIC(I,1)                                                     GRA08400
      MVTDIR(I)=IDIR(I,1)                                               GRA08410
      SPD(I)=AVSP                                                       GRA08420
      IUSTAT(I)=0                                                       GRA08430
      IPRDIR(I)=IDIR(I,1)                                               GRA08440
      ISECWD(I)=120                                                     GRA08450
      NF(I)=1                                                           GRA08460
      II(I)=1                                                           GRA08470
    3 CONTINUE                                                          GRA08480
C                                                                       GRA08490
C   READ IN DEFENSE UNIT'S LOCATIONS                                    GRA08500
C                                                                       GRA08510
      SUMBO=0.0                                                         GRA08520
      DO 4 I=K,L                                                        GRA08530
      READ(9,104) X(I),Y(I),FL(I),IPRDIR(I),ISECWD(I)                   GRA08540
  104 FORMAT(F6.1,1X,F6.1,1X,F3.1,1X,I3,1X,I3)                          GRA08550
      FO(I)=FL(I)                                                       GRA08560
      SUMBO=SUMBO+FO(I)                                                 GRA08570
```

```
      MVTDIR(I)=0                                                       GRA08580
      SPD(I)=0.0                                                        GRA08590
      IUSTAT(I)=0                                                       GRA08600
      IMOVE(I)=0                                                        GRA08610
    4 CONTINUE                                                          GRA08620
C                                                                       GRA08630
C CHECK FOR ALTERNATE DEFENSE POSITIONS AND READ IN IF WANTED           GRA08640
C                                                                       GRA08650
      READ(9,400) IALT,BREAK,ITEM                                       GRA08660
  400 FORMAT(I1,1X,F6.1,1X,I2)                                          GRA08670
      IF(IALT.EQ.1) GO TO 401                                           GRA08680
      DO 402 I=K,L                                                      GRA08690
      READ(9,107) XA(I),YA(I)                                           GRA08700
  402 CONTINUE                                                          GRA08710
  401 DELT=10.                                                          GRA08720
      TA(1)=20.                                                         GRA08730
      T1(1)=8.                                                          GRA08740
      TH(1)=8.                                                          GRA08750
      TM(1)=10.                                                         GRA08760
      TF1(1)=1.                                                         GRA08770
      TF2(1)=1.                                                         GRA08780
      TF3(1)=1.                                                         GRA08790
      TA(2)=20.                                                         GRA08800
      T1(2)=8.                                                          GRA08810
      TH(2)=8.                                                          GRA08820
      TM(2)=15.                                                         GRA08830
      TF1(2)=10.                                                        GRA08840
      TF2(2)=12.                                                        GRA08850
      TF3(2)=15.                                                        GRA08860
C                                                                       GRA08870
C   READ IN HIT AND KILL PROBABILITIES                                  GRA08880
C                                                                       GRA08890
      DO 5 I=1,2                                                        GRA08900
      DO 514 J=1,6                                                      GRA08910
      READ(9,515) P(I,J),PHH(I,J),PHM(I,J),PKH(I,J)                     GRA08920
  515 FORMAT(4(F4.2,1X))                                                GRA08930
  514 CONTINUE                                                          GRA08940
    5 CONTINUE                                                          GRA08950
      PTT(1,1)=1.0                                                      GRA08960
      PTT(1,2)=0.8                                                      GRA08970
      PTT(2,2)=0.2                                                      GRA08980
      PTT(1,3)=0.8                                                      GRA08990
      PTT(2,3)=0.15                                                     GRA09000
      PTT(3,3)=0.05                                                     GRA09010
      DO 31 I=1,NRU                                                     GRA09020
      DO 31 J=K,L                                                       GRA09030
      NLOSC(I,J)=0                                                      GRA09040
      NLOSC(J,I)=0                                                      GRA09050
```

```
      Q(I,J)=1.0
      Q(J,I)=1.0
      VISFR(I,J)=0.0
      VISFR(J,I)=0.0
   31 CONTINUE
      IC=1
C
C  PRINT INITIAL BATTLE INFORMATION
C
      WRITE(6,599)
  599 FORMAT('1',1X,'INITIAL GROUND COMBAT INFORMATION')
      WRITE(6,600)
  600 FORMAT(/1X,'UNIT',7X,'X',8X,'Y',4X,'FORCE LEVEL')
      DO 601 I=1,L
      WRITE(6,602) I,X(I),Y(I),FL(I)
  602 FORMAT(1X,I3,3X,F7.1,2X,F7.1,7X,F3.1)
  601 CONTINUE
      IF(ITRIT.EQ.1) GO TO 603
      WRITE(6,604)
  604 FORMAT(/1X,'ATTRITION IS STOCHASTIC'/)
      GO TO 605
  603 WRITE(6,606)
  606 FORMAT(/1X,'ATTRITION IS DETERMINISTIC'/)
  605 IF(IRTE.EQ.0) GO TO 607
      WRITE(6,608)
  608 FCRMAT(1X,'ROUTES DETERMINED BY USER'/)
  607 WRITE(6,609) AVSP
  609 FORMAT(1X,'ATTACK VEHICLE SPEED IS ',F4.1/)
      WRITE(6,610) BREAK
  610 FORMAT(1X,'BREAKPOINT DISTANCE IS ',F6.1/)
      IF(IALT.EQ.0) GO TO 615
      WRITE(6,620)
  620 FORMAT(1X,'DEFENDER WILL NOT MOVE TO ALTERNATE POSITIONS'/)
      GO TO 625
  615 WRITE(6,630)
  630 FORMAT(1X,'DEFENDER WILL MOVE TO ALTERNATE POSITIONS'/1X,
     1'ALTERNATE POSITIONS ARE:'/1X,'UNIT',5X,'X',8X,'Y')
      DO 635 I=K,L
      WRITE(6,640) I,XA(I),YA(I)
  640 FORMAT(1X,I3,3X,F7.1,2X,F7.1)
  635 CONTINUE
  625 IRAN=500
      WRITE(6,645)
  645 FORMAT(/4X,'ATK KILL PROBABILITIES'/1X,'RANGE',4X,'P',
     14X,'PHH',3X,'PHM',3X,'PKH')
      DO 650 I=1,6
      WRITE(6,655) IRAN,P(1,I),PHH(1,I),PHM(1,I),PKH(1,I)
  655 FORMAT(2X,I4,4(2X,F4.2))
```

```
      IRAN=IRAN+500                                                     GRA09540
  650 CONTINUE                                                          GRA09550
      IRAN=500                                                          GRA09560
      WRITE(6,660)                                                      GRA09570
  660 FORMAT(/4X,'DEF. KILL PROBABILITIES'/1X,'RANGE',4X,'P',           GRA09580
     14X,'PHH',3X,'PHM',3X,'PKH')                                       GRA09590
      DO 665 I=1,6                                                      GRA09600
      WRITE(6,655) IRAN,P(2,I),PHH(2,I),PHM(2,I),PKH(2,I)               GRA09610
      IRAN=IRAN+500                                                     GRA09620
  665 CONTINUE                                                          GRA09630
      WRITE(6,670)                                                      GRA09640
670   FORMAT('1',10X,'BATTLE BEGINS',//)                                GRA09650
C     UPDATE LOCATION OF RED UNITS.                                     GRA09660
C                                                                       GRA09670
      DISMAX=5000.0                                                     GRA09680
   67 DO 9 I=1,NRU                                                      GRA09690
      IF(IUSTAT(I).EQ.2) GOTO 9                                         GRA09700
      IF(IUSTAT(I).EQ.0) GOTO 76                                        GRA09710
      NF(I)=NF(I)+1                                                     GRA09720
      IF(NF(I).LT.NOD) GOTO 9                                           GRA09730
      NF(I)=1                                                           GRA09740
   76 DO 11 J = 1, NRU                                                  GRA09750
      IF (J .EQ. I) GO TO 11                                            GRA09760
      IF (IUSTAT(J) .EQ. 2) GO TO 11                                    GRA09770
      DIST = X(I) - X(J)                                                GRA09780
      IF (DIST .GT. DISMAX) GO TO 9                                     GRA09790
   11 CONTINUE                                                          GRA09800
      II(I) = II(I) + 1                                                 GRA09810
      K7=II(I)                                                          GRA09820
      X(I)=XIC(I,K7)                                                    GRA09830
      Y(I)=YIC(I,K7)                                                    GRA09840
      MVTDIR(I)=IDIR(I,K7)                                              GRA09850
      IPRDIR(I)=IDIR(I,K7)                                              GRA09860
C     WRITE(6,666) I,X(I),Y(I),MVTDIR(I),IPRDIR(I)                      GRA09870
C 666 FORMAT(/,1X,I3,1X,F10.5,2X,F10.5,2X,I10,2X,I10,/)                 GRA09880
    9 CONTINUE                                                          GRA09890
C                                                                       GRA09900
C     LINE--OF-SIGHT CHECK BETWEEN UNITS AND TARGETS SELECTION          GRA09910
C                                                                       GRA09920
      DO 17 J=K,L                                                       GRA09930
      NT(J)=0                                                           GRA09940
   17 CONTINUE                                                          GRA09950
      DO 12 I=1,NRU                                                     GRA09960
      NT(I)=0                                                           GRA09970
      IF(IUSTAT(I).EQ.2) GOTO 12                                        GRA09980
      DO 16 J=K,L                                                       GRA09990
      IF(IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 16                     GRA10000
      XX1=X(I)                                                          GRA10010
```

```
      YY1=Y(I)                                                           GRA10020
  133 CALL ELEV(XX1,YY1,TMACI)                                           GRA10030
      XX2=X(J)                                                           GRA10040
      YY2=Y(J)                                                           GRA10050
      CALL ELEV(XX2,YY2,TMACJ)                                           GRA10060
      LATOB=1                                                            GRA10070
      LBTOA=1                                                            GRA10080
C     WRITE(6,675) XX1,YY1,TMACI,XX2,YY2,TAMCJ                           GRA10090
C 675 FORMAT(1X,'PRELOS',1X,6(F10.5,1X))                                 GRA10100
      CALL LOS(XX1,YY1,TMACI,0.0,SIZETK,XX2,YY2,TMACJ,0.0,SIZETW,        GRA10110
     1LATOB,LBTOA,VISFRA,VISFRB)                                         GRA10120
      VISFR(I,J)=VISFRA                                                  GRA10130
      VISFR(J,I)=VISFRB                                                  GRA10140
      IF(VISFRA.GT.ZL) GOTO 18                                           GRA10150
      LOST(I,J)=0                                                        GRA10160
      LOST(J,I)=0                                                        GRA10170
      NLOSC(I,J)=NLOSC(I,J)+1                                            GRA10180
      NLOSC(J,I)=NLOSC(I,J)                                              GRA10190
      GOTO 16                                                            GRA10200
   18 LOST(I,J)=1                                                        GRA10210
      LOST(J,I)=1                                                        GRA10220
      NLOSC(I,J)=0                                                       GRA10230
      NLOSC(J,I)=0                                                       GRA10240
      RANGE=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)                          GRA10250
      IF(RANGE.LT.RMINTK.OR.RANGE.GT.RMXTK)GOTO 20                       GRA10260
      IF(Q(I,J).EQ.1.0) GOTO 20                                          GRA10270
      IUSTAT(I)=1                                                        GRA10280
      NT(I)=NT(I)+1                                                      GRA10290
      M=NT(I)                                                            GRA10300
      LOT(I,M)=J                                                         GRA10310
      RCT(I,M)=RANGE                                                     GRA10320
      IF(M.EQ.1) GOTO 20                                                 GRA10330
      CALL SORT(I,M)                                                     GRA10340
   20 IF(RANGE.LT.RMINTW.OR.RANGE.GT.RMXTW) GOTO 16                      GRA10350
      IF(Q(J,I).EQ.1.0) GOTO 16                                          GRA10360
      IUSTAT(J)=1                                                        GRA10370
      NT(J)=NT(J)+1                                                      GRA10380
      M=NT(J)                                                            GRA10390
      LOT(J,M)=I                                                         GRA10400
      RCT(J,M)=RANGE                                                     GRA10410
      IF(M.EQ.1) GOTO 16                                                 GRA10420
      CALL SORT(J,M)                                                     GRA10430
   16 CONTINUE                                                           GRA10440
   12 CONTINUE                                                           GRA10450
      DO 25 I=1,NRU                                                      GRA10460
      IF(IUSTAT(I).EQ.2) GOTO 25                                         GRA10470
      IF(NT(I).NE.0) GOTO 25                                             GRA10480
      IUSTAT(I)=0                                                        GRA10490
```

```
      NF(I)=0
   25 CONTINUE
      DO 79 J=K,L
      IF(IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 79
      IF(NT(J).EQ.0) IUSTAT(J)=0
   79 CONTINUE
C
C     UPDATE OF THE ACCUMULATED DETECTION PROBABILITIES.
C
      IAA=1
      IBB=NRU
      ICC=K
      IDD=L
      FR=TOWFR
      OP=PM
      DO 345 I=1,6
  345 CONTINUE
   37 DO 14 I=IAA,IBB
      IF(IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 14
      DO 19 J=ICC,IDD
      PROP=0.0
      IF(IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 19
      OLDQ(I,J)=Q(I,J)
      IF(LOST(I,J).EQ.0) GOTO 15
      IF(NT(I).GT.0) GOTO 22
      PCTVIS=VISFR(J,I)
      CALL LAMDA(I,J,PCTVIS,DETRAT,PSUBK)
      QV=EXP(-(FL(I)*DETRAT*OP*DELT*FL(J)))
      IF(NT(J).GT.0) GOTO 23
      Q(I,J)=Q(I,J)*QV
      GOTO 19
   23 QP=(1.0-PSUBK)**(FR*DELT*FL(I)*FL(J))
      Q(I,J)=Q(I,J)*(QV+QP-QV*QP)
      GOTO 19
   22 N5=NT(I)
      DO 24 I1=1,N5
      K1=LOT(I,I1)
      ANG1=ATAN2(Y(K1)-Y(I),X(K1)-X(I))
      ANG2=ATAN2(Y(J)-Y(I),X(J)-X(I))
      IF((ANG1*ANG2).GE.0.0) GOTO 77
      IF(ANG2.LT.0.0) GOTO 32
      ANG=2*PAI+ANG1-ANG2
      GOTO 35
   32 ANG=2*PAI+ANG2-ANG1
   35 IF(ANG.GT.PAI) ANG=2*PAI-ANG
      GOTO 33
   77 ANG=ABS(ANG2-ANG1)
   33 AA=15.0*PAI/180.0
```

```
      IF(ANG.GT.AA) GOTO 24                                           GRA10980
      PROP=PROP+PTT(I1,N5)                                            GRA10990
   24 CONTINUE                                                        GRA11000
      IF(PROP.EQ.0.0) GOTO 34                                         GRA11010
      IF(NT(J).GT.0)GOTO 36                                           GRA11020
      CALL LAMDA(I,J,PCTVIS,DETRAT,PSUBK)                             GRA11030
      DETRAT=DETRAT*RF                                                GRA11040
      QV=EXP(-(FL(I)*PROP*DETRAT*DELT*FL(J)))                         GRA11050
      Q(I,J)=Q(I,J)*QV                                                GRA11060
      GOTO 19                                                         GRA11070
   36 Q(I,J)=0.0                                                      GRA11080
      GOTO 19                                                         GRA11090
   34 IF(IAA.EQ.1) GOTO 19                                            GRA11100
      Q(I,J)=1.0                                                      GRA11110
      GOTO 19                                                         GRA11120
   15 IF(NLOSC(I,J).LE.3) GOTO 19                                     GRA11130
      Q(I,J)=1.0                                                      GRA11140
   19 CONTINUE                                                        GRA11150
   14 CONTINUE                                                        GRA11160
      IF(IAA.EQ.K) GOTO 38                                            GRA11170
      FR=TNKFR                                                        GRA11180
      IAA=K                                                           GRA11190
      IBB=L                                                           GRA11200
      ICC=1                                                           GRA11210
      IDD=NRU                                                         GRA11220
      OP=1.0                                                          GRA11230
      GOTO 37                                                         GRA11240
C                                                                     GRA11250
C     FIRE ALLOCATION.                                                GRA11260
C                                                                     GRA11270
   38 DO 28 I=1,L                                                     GRA11280
   28 NA(I)=0                                                         GRA11290
      DO 26 I=1,L                                                     GRA11300
      IF(IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 26                   GRA11310
      IF(NT(I).EQ.0) GOTO 26                                          GRA11320
      DO 27 J=1,3                                                     GRA11330
      APOA(I,J)=0.0                                                   GRA11340
   27 CONTINUE                                                        GRA11350
      IF(NT(I).EQ.1) GOTO 78                                          GRA11360
      IF(NT(I).EQ.2) GOTO 29                                          GRA11370
      NOT=3                                                           GRA11380
      MM1=LOT(I,1)                                                    GRA11390
      MM2=LOT(I,2)                                                    GRA11400
      MM3=LOT(I,3)                                                    GRA11410
      PROB=(1.0-Q(I,MM1))*Q(I,MM2)*Q(I,MM3)                           GRA11420
      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB                               GRA11430
      PROB=Q(I,MM1)*(1.0-Q(I,MM2))*Q(I,MM3)                           GRA11440
      APOA(I,2)=APOA(I,2)+PTT(1,1)*PROB                               GRA11450
```

```
      PROB=Q(I,MM1)*Q(I,MM2)*(1.0-Q(I,MM3))
      APOA(I,3)=APOA(I,3)+PTT(1,1)*PROB
      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*Q(I,MM3)
      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
      APOA(I,2)=APOA(I,2)+PTT(2,2)*PROB
      PROB=(1.0-Q(I,MM1))*Q(I,MM2)*(1.0-Q(I,MM3))
      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
      APOA(I,3)=APOA(I,3)+PTT(2,2)*PROB
      PROB=Q(I,MM1)*(1.0-Q(I,MM2))*(1.0-Q(I,MM3))
      APOA(I,2)=APOA(I,2)+PTT(1,2)*PROB
      APOA(I,3)=APOA(I,3)+PTT(2,2)*PROB
      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*(1.0-Q(I,MM3))
      APOA(I,1)=APOA(I,1)+PTT(1,3)*PROB
      APOA(I,2)=APOA(I,2)+PTT(2,3)*PROB
      APOA(I,3)=APOA(I,3)+PTT(3,3)*PROB
   30 DO 44 J=1,NOT
      KK=LOT(I,J)
      NA(KK)=NA(KK)+1
      IN=NA(KK)
      LOA(KK,IN)=I
      POA(KK,IN)=APOA(I,J)
   44 CONTINUE
      GOTO 26
   29 NOT=2
      MM1=LOT(I,1)
      MM2=LOT(I,2)
      PROB=(1.0-Q(I,MM1))*Q(I,MM2)
      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB
      PROB=Q(I,MM1)*(1.0-Q(I,MM2))
      APOA(I,2)=APOA(I,2)+PTT(1,1)*PROB
      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))
      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
      APOA(I,2)=APOA(I,2)+PTT(2,2)*PROB
      GOTO 30
   78 NOT=1
      MM1=LOT(I,1)
      PROB=1.0-Q(I,MM1)
      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB
      GOTO 30
   26 CONTINUE
C
C     ATTRITION.
C
C
      SUMR=0.0
      SUMB=0.0
      DO 40 I=1,L
      IF(IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 40
```

```
      M6=NA(I)
      SUM=0.0
      IF(M6.EQ.0) GOTO 47
      DO 41 J=1,M6
      M7=LOA(I,J)
      IF(M7.LT.K) GOTO 42
      ITYPE=2
      GOTO 43
   42 ITYPE=1
   43 RANGE=SQRT((X(I)-X(M7))**2+(Y(I)-Y(M7))**2)
      IF (ITRIT.EQ.1) GO TO 131
      CALL STOCH(ITYPE,RANGE,AJI)
      GO TO 5000
  131 CALL ETK(ITYPE,RANGE,T)
      AJI=1.0/T
 5000 SUM=SUM+AJI*FL(M7)*POA(I,J)*DELT
   41 CONTINUE
   47 OFL(I)=FL(I)
      FL(I)=FL(I)-SUM
      IF(FL(I).GT.ZL) GOTO 46
      FL(I)=0.0
      IUSTAT(I)=2
   46 IF(I.LT.K) GOTO 60
      SUMB=SUMB+FL(I)
      TPOL(I)=(FO(I)-FL(I))/FO(I)
      GO TO 40
   60 SUMR=SUMR+FL(I)
      TPOL(I)=(FO(I)-FL(I))/FO(I)
   40 CONTINUE
C
C     PRINT AND CHECK FOR BATTLE DETERMINATION.
C
C
      ITIME=IC*10
      DO 57 I=K,L
      IF(IUSTAT(I).EQ.2) GO TO 57
      DO 58 J=1,NRU
      IF(IUSTAT(J).EQ.2) GO TO 58
      CHECK=X(I)-X(J)
      AVD=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
      IF(AVD.LT.BREAK.OR.CHECK.LT.50.) GO TO 250
   58 CONTINUE
   57 CONTINUE
      GO TO 99
C
C    COMPLETE ATTACK UNIT'S MOVE
C
  250 DO 251 I=K,L
```

```
      IF(IALT.EQ.1.OR.IMOVE(I).EQ.ITEM) GO TO 6000
      IF(IUSTAT(I).EQ.0) IUSTAT(I)=3
      IMOVE(I)=IMOVE(I)+1
      IF(IMOVE(I).LT.ITEM) GO TO 251
      X(I)=XA(I)
      Y(I)=YA(I)
      IF(IUSTAT(I).EQ.3) IUSTAT(I)=0
  251 CONTINUE
   99 IITIME=ITIME+IFIX(GATM)
      WRITE(6,112) IITIME
  112 FORMAT(////1X,'TIME=',I4,1X,'SECONDS'//)
      WRITE(6,113)
  113 FORMAT(1X,'UNIT',5X,'X',8X,'Y',5X,'FORCE LEVEL',2X,'STATUS',
     12X,'LOST-PCT',2X,'TARGETS')
      DO 59 I=1,L
      N6=NT(I)
      IF(N6.NE.0) GO TO 48
      WRITE(6,264) I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I)
  264 FORMAT(3X,I1,3X,F7.1,2X,F7.1,6X,F3.1,9X,I1,6X,F5.3)
      GO TO 59
   48 WRITE(6,114)I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I),
     1(LOT(I,J),J=1,N6)
  114 FORMAT(3X,I1,3X,F7.1,2X,F7.1,6X,F3.1,9X,I1,6X,F5.3,3X,3(I1,1X))
   59 CONTINUE
C
C     CHECK FOR BATTLE TERMINATION.
C
      IOT=0
      DO 53 I=1,NRU
      IF(FL(I).EQ.0.0) GOTO 53
      IOT=1
   53 CONTINUE
      IF(IOT.EQ.1) GOTO 54
      WRITE(6,117)
  117 FORMAT(1X,'*ATTACK FORCE IS ELIMINATED.  END OF BATTLE.')
      GOTO 66
   54 IOT=0
      DO 55 I=K,L
      IF(FL(I).EQ.0.0) GOTO 55
      IOT=1
   55 CONTINUE
      IF(IOT.EQ.1) GOTO 65
      WRITE(6,118)
  118 FORMAT(1X,'*DEFENSE FORCE IS ELIMINATED. END OF BATTLE.')
      GOTO 66
 6000 WRITE (6,119)
  119 FORMAT(1X,'* DISTANCE BETWEEN FORCES IS TOO CLOSE. END OF BATTLE
     1')
```

```
      GO TO 66
   65 IC=IC+1
      GO TO 67
   66 RETURN
      END
C
      SUBROUTINE SETUP
C  SUBROUTINE SETUP IS USED TO READ IN THE TERRAIN DATA AND
C  CREATE PARAMETRIC TERRAIN. THIS TERRAIN DATA WILL BE USED
C  WHEN COMPUTING LINE-OF-SIGHT BETWEEN TARGETS AND OBSERVERS
C  AS WELL AS PROVIDING A GRID SYSTEM FCR UNIT LOCATIONS AND
C  MOVEMENT.
C
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100),SPRD(100)
      COMMON /HILLS/ ECC(100),PXX(100),PYY(100),PXY(100),BASE
      COMMON /HILLS/ NHILLS
      COMMON /COVER/ CXC(150),CYC(150),CPEAK(150),CPXX(150),CPYY(150)
      COMMON /COVER/ CPXY(150),NCVELS
      COMMON /COUNTR/KH,KHW,KV,KN,KGRS,KELL,KINT
      CCMMON /GRID/ LST(5,4),NHL(5,4),LISTH(150),KHREP(150),KTREP
      COMMON /GRID/ LSTC(5,4),NC(5,4),LISTC(400),KCREP(150)
      PAI=3.14159
      L = 5
      READ(L,7) NHILLS
      READ(L,47) BASE
47    FORMAT(F10.4)
7     FORMAT(I6)
17    FORMAT(6F10.3)
      DO 50 I=1,NHILLS
      READ(L,17) XC(I),YC(I),PEAK(I),ANGH(I),SPRD(I),ECC(I)
50    CONTINUE
      READ(L,37)LST
      READ(L,37)NHL
      READ(L,7)NHTOT
      READ(L,37)(LISTH(I),I=1,NHTOT)
37    FORMAT(10I5)
   65 DO 100 I=1,NHILLS
      AANGLE=ANGH(I)*PAI/180.
      SANG=SIN(AANGLE)
      CANG=COS(AANGLE)
      A=PEAK(I)/(PEAK(I)-50.)
      A=ALOG(A)
      B=A*ECC(I)**2
      SSPD=SPRD(I)**2
      PXX(I)=-(A*CANG*CANG+B*SANG*SANG)/SSPD
      PYY(I)=-(A*SANG*SANG+B*CANG*CANG)/SSPD
      PXY(I)=(2.*SANG*CANG*(B-A))/SSPD
      KHREP(I)=-2147483600
```

```
C  ALL VALUES NOW IN METERS ON 0 -- 10,000 GRID
100    CONTINUE
       READ(L,7) NCVELS
       IF(NCVELS.EQ.0) GO TO 75
       DO 60 I=1,NCVELS
       READ(L,27)CXC(I),CYC(I),CPEAK(I),CPXX(I),CPYY(I),CPXY(I)
27     FORMAT(3F10.4,3E13.7)
       KCREP(I)=-2147483600
60     CONTINUE
       READ(L,37)LSTC
       READ(L,37)NC
       READ(L,7)NCTOT
       READ(L,37)(LISTC(I),I=1,NCTOT)
    75 KTREP=-2147483600
       KH=0
       KHW=0
       KV=0
       KN=0
       KGRS=0
       KELL=0
       KINT=0
       RETURN
       END
C
       SUBROUTINE ROUTE
C
C  SUBROUTINE ROUTE COMPUTES THE ROUTE OF EACH ATTACKING  UNIT
C  WHEN THE USER HAS SELECTED THE OPTION OF INPUTING ATTACKER
C  ROUTES. IT CALCULATES THE COORDINATES OF EACH INTERVAL ENDPOINT
C  ALONG THE ROUTE,MAKING EACH INTERVAL LENGTH(DISTANCE MOVED DURING
C  A 10 SECOND TIME STEP) THE SAME. THE INTERVAL LENGTH IS DETERMINED
C  BY THE SPEED THE USER HAS SELECTED AND INPUTED FOR THE CURRENT
C  BATTLE.
C
       COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),
      1ICIR(3,200),AVSP,ISPD
      1,IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZETK,
      1SIZETW,NT(6),NF(6),SRF,DISMAX,
      1NLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,
      1PTT(3,3),RF,POA(6,6),APOA(6,6),LOA(6,6),NA(6),OFL(6),POL(6)
       DIMENSION XLOC(3,20),YLOC(3,20),N(3)
       IF(ISPD.EQ.4) DST=80.463
       IF(ISPD.EQ.3) DST=67.053
       IF(ISPD.EQ.2) DST=53.643
       IF(ISPD.EQ.1) DST=40.232
       LN=9
       DO 300 I=1,NRU
       READ(LN,15) N(I)
```

```
 15 FORMAT(I2)                                                    GRA13860
    NL=N(I)+1                                                     GRA13870
    DO 200 IN=2,NL                                                GRA13880
    READ(LN,201) XLOCS,YLOCS                                      GRA13890
201 FORMAT(F6.1,1X,F6.1)                                          GRA13900
    XLOC(I,IN)=XLOCS                                              GRA13910
    YLOC(I,IN)=YLOCS                                              GRA13920
200 CONTINUE                                                      GRA13930
    XLOC(I,1)=XIC(I,1)                                            GRA13940
    YLOC(I,1)=YIC(I,1)                                            GRA13950
    IDIR(I,1)=0                                                   GRA13960
    NL=N(I)                                                       GRA13970
    NUM=2                                                         GRA13980
    DC 305 J=1,NL                                                 GRA13990
    XL=XLOC(I,J+1)-XLOC(I,J)                                      GRA14000
    YL=YLOC(I,J+1)-YLOC(I,J)                                      GRA14010
    DIST=SQRT(XL**2+YL**2)                                        GRA14020
    Y=ABS(YL)                                                     GRA14030
    Z=Y/XL                                                        GRA14040
    ANGL=ATAN(Z)                                                  GRA14050
    DEG=ANGL*57.2958                                              GRA14060
    IF(J.EQ.1) GO TO 320                                          GRA14070
    XLN=(DST-EXTRA)*COS(ANGL)                                     GRA14080
    DIST=(DIST+EXTRA)-DST                                         GRA14090
    YLN=(DST-EXTRA)*SIN(ANGL)                                     GRA14100
    XIC(I,NUM)=XIC(I,NUM-1)+XLN+XLE                               GRA14110
    IF(YL.GT.0.) GO TO 325                                        GRA14120
    YLN=-YLN                                                      GRA14130
325 YIC(I,NUM)=YIC(I,NUM-1)+YLN+YLE                               GRA14140
    IF(YL.GT.0.) GO TO 340                                        GRA14150
    IDIR(I,NUM)=-IFIX(DEG)                                        GRA14160
    GO TO 341                                                     GRA14170
340 IDIR(I,NUM)=IFIX(DEG)                                         GRA14180
341 NUM=NUM+1                                                     GRA14190
320 XLN=DST*COS(ANGL)                                             GRA14200
    YLN=DST*SIN(ANGL)                                             GRA14210
    IF(YL.GT.0.) GO TO 310                                        GRA14220
    YLN=-YLN                                                      GRA14230
310 IF(DIST.LT.DST) GO TO 315                                     GRA14240
    XIC(I,NUM)=XIC(I,NUM-1)+XLN                                   GRA14250
    YIC(I,NUM)=YIC(I,NUM-1)+YLN                                   GRA14260
    IF(YL.GT.0.) GO TO 342                                        GRA14270
    IDIR(I,NUM)=-IFIX(DEG)                                        GRA14280
    GO TO 343                                                     GRA14290
342 IDIR(I,NUM)=IFIX(DEG)                                         GRA14300
343 DIST=DIST-DST                                                 GRA14310
    NUM=NUM+1                                                     GRA14320
    GO TO 310                                                     GRA14330
```

```
  315 EXTRA=DIST                                                        GRA14340
      XLE=EXTRA*COS(ANGL)                                               GRA14350
      YLE=EXTRA*SIN(ANGL)                                               GRA14360
      IF(YL.GT.0.) GO TO 305                                            GRA14370
      YLE=-YLE                                                          GRA14380
  305 CONTINUE                                                          GRA14390
  300 CONTINUE                                                          GRA14400
      RETURN                                                            GRA14410
      END                                                               GRA14420
C                                                                       GRA14430
      SUBROUTINE LAMDA(I,J,PCTVIS,DETRAT,PK)                            GRA14440
C                                                                       GRA14450
C   SUBROUTINE LAMDA IN CONJUCTION WITH THE LOS ROUTINE COMPUTES        GRA14460
C   THE DETECTION RATE(DETRAT) OF TARGET J BY THE OBSERVER I GIVEN      GRA14470
C   THE PERCENT OF TARGET VISIBLE (PCTVIS) TO THE OBSERVER.             GRA14480
C                                                                       GRA14490
      CCMMON /GRP1/ IPRDIR(6),ISECWD(6),MVTDIR(6),X(6),Y(6),SPD(6)      GRA14500
      TCFACT=1.0                                                        GRA14510
      ZEROL=0.00001                                                     GRA14520
      PAI=3.14159                                                       GRA14530
    7 D=(ISECWD(I)*PAI/180.0)/2.0                                       GRA14540
      BBB=(1.0/(2.0*(SIN(D)-D*COS(D))))                                 GRA14550
      IF(ABS(BBB).LT.ZEROL) BBB=0.0                                     GRA14560
      AAA=(-BBB)*COS(D)                                                 GRA14570
      IF(ABS(AAA).LT.ZEROL) AAA=0.0                                     GRA14580
      OTANG=ATAN2((Y(J)-Y(I)),(X(J)-X(I)))                              GRA14590
      IF(OTANG.LT.-PAI/2.AND.OTANG.GT.-PAI) OTANG=2*PAI+OTANG           GRA14600
      PD=IPRDIR(I)*PAI/180.0                                            GRA14610
      IF((PD*OTANG).GE.0.0) GOTO 1                                      GRA14620
      IF(PD.LT.0.0) GOTO 9                                              GRA14630
      ANGLE=2*PAI+OTANG-PD                                              GRA14640
      GOTO 10                                                           GRA14650
    9 ANGLE=2*PAI+PD-OTANG                                              GRA14660
   10 IF(ANGLE.GT.PAI) ANGLE=2*PAI-ANGLE                                GRA14670
      GOTO 2                                                            GRA14680
    1 ANGLE=ABS(PD-OTANG)                                               GRA14690
    2 IF(ANGLE.GT.D) GOTO 3                                             GRA14700
      DUP=PD+D                                                          GRA14710
      DLOW=PD-D                                                         GRA14720
      ANGLFT=OTANG+(15.0*PAI/180.)                                      GRA14730
      IF(ANGLFT.GT.DUP) ANGLFT=DUP                                      GRA14740
      ANGLRT=OTANG-(15.*PAI/180.)                                       GRA14750
      IF(ANGLRT.LT.DLOW) ANGLRT=DLOW                                    GRA14760
      PK=BBB*ABS(ABS(SIN(ANGLFT))-ABS(SIN(ANGLRT)))+AAA*(ANGLFT-        GRA14770
     1ANGLRT)                                                           GRA14780
      IF(PK.LT.0.0) GOTO 3                                              GRA14790
      IF(PK.GT.1.0) GOTO 5                                              GRA14800
      GOTO 8                                                            GRA14810
```

```
    3 PK=0.0
      DETRAT=0.0
      GOTO 6
    5 PK=1.0
    8 RANGE=SQRT((X(J)-X(I))**2+(Y(J)-Y(I))**2)
      RR=0.001*RANGE/PCTVIS
      TOANG=ATAN2((Y(I)-Y(J)),(X(I)-X(J)))
      AD=MVTDIR(J)*PAI/180.0
      HORVEL=ABS(SPD(J)*SIN(TOANG-AD))
      HORVEL=HORVEL*1609.3/3600.0
      DENOM=1.453+TCFACT*(0.5978+2.188*(RR**2)-0.5038*HORVEL)
      IF(DENOM.LE.ZEROL) DENOM=ZEROL
      DETRAT=0.003+1.088/DENOM
      DETRAT=DETRAT*PK
    6 RETURN
      END
C
      SUBROUTINE ELEV(X,Y,TMAC)
C
C   SUBROUTINE LEV DETERMINES THE TERRAIN ELEVATION FOR A GIVEN
C   SET OF X, Y COORDINATES. THIS FUNCTION IS USED IN CONJUNCTION
C   WITH THE LOS SUBROUTINE IN COMPUTING LINE-OF-SIGHT BETWEEN
C   OBSERVER AND TARGET.
C
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100),SPRD(100)
      COMMON /HILLS/ ECC(100),PXX(100),PYY(100),PXY(100),BASE
      COMMON /HILLS/ NHILLS
      COMMON /GRID/ LST(5,4),NHL(5,4),LISTH(150),KHREP(150),KTREP
      COMMON /GRID/ LSTC(5,4),NC(5,4),LISTC(400),KCREP(150)
      DATA GSIZE/1000./
C FUNCTION TO COMPUTE TERRAIN ELEVATION FOR GIVEN X, Y COORDINATES.
      ZMAX=BASE
      IX=1+IFIX(X/GSIZE)
      IY=1+IFIX(Y/GSIZE)
      IF(NHL(IX,IY).EQ.0) GO TO 150
      LS=LST(IX,IY)
      LEND=LS+NHL(IX,IY)-1
      DO 100 L=LS,LEND
      I=LISTH(L)
      QX=X-XC(I)
      QY=Y-YC(I)
      QXSQ=QX*QX
      QYSQ=QY*QY
      QXY=QX*QY
      FACTOR=PXX(I)*QXSQ+PYY(I)*QYSQ+PXY(I)*QXY
      IF(FACTOR.LT.-3.) GO TO 100
      HT=PEAK(I)*EXP(FACTOR)
      IF(HT.LE.ZMAX) GO TO 100
```

```
      ZMAX=HT
  100 CONTINUE
  150 TMAC=ZMAX
      RETURN
      END
C
      SUBROUTINE STOCH(I,RANGE,A)
C
C  SUBROUTINE STOCH DETERMINES THE ATTRITION COEFFICIENTS WHEN
C  A USER HAS SELECTED A STOCHASTIC ATTRITION OPTION. THE CALCULATION
C  IS A FUNCTION OF THE ORIGINAL STOCHASTICALLY DETERMINED ATTRITION
C  COEFFICIENT AS WELL AS A FUNCTION OF RANGE.
C
      COMMON /GRP6/ ALPHA(6)
      COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),
     1IDIR(3,200),AVSP,ISPD
     1,IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZETK,
     1SIZETW,NT(6),NF(6),SRF,DISMAX,
     1NLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,
     1PTT(3,3),RF,POA(6,6),APOA(6,6),LOA(6,6),NA(6),OFL(6),POL(6)
      IF(I.EQ.2) GO TO 5003
      A=ALPHA(I)*((1.0-RANGE/RMXTW)**2)
      GO TO 5004
 5003 A=ALPHA(I)*((1.0-RANGE/RMXTK)**2)
 5004 RETURN
        END
C
      SUBROUTINE ETK(I,RANGE,T)
C
C  SUBROUTINE ETK COMPUTES THE EXPECTED TIME FOR A GIVEN FIRER TO
C  KILL A GIVEN TARGET. THE CALCULATION IS A FUNCTION OF RANGE,
C  TIME OF FLIGHT FOR A ROUND AND HIT AND KILL PROBABILITIES FOR
C  THE FIRING WEAPON SYSTEM. IT IS A NUMBER THAT IS USED IN THE
C  CCMPUTATION OF THE DETERMINISTIC ATTRITION COEFFICIENTS.
C
      CCMMON /GRP2/ TA(2),T1(2),TH(2),TM(2),TF1(2),TF2(2),TF3(2),
     1P(2,6),PHH(2,6),PHM(2,6),PKH(2,6),TF(2)
      IF(I.EQ.2) GOTO 5
      TF(I)=TF1(I)
      GOTO 6
    5 IF(RANGE.GT.1000.0) GOTO 7
      TF(I)=TF1(I)-(TF1(I)*(1000.0-RANGE)/1000.0)
      GOTO 6
    7 IF(RANGE.GT.2000.0) GOTO 8
      TF(I)=TF2(I)-((TF2(I)-TF1(I))*(2000.0-RANGE)/1000.0)
      GOTO 6
    8 TF(I)=TF3(I)-((TF3(I)-TF2(I))*(3000.0-RANGE)/1000.0)
    6 J=(RANGE+250.0)/500.0
```

```
      IF(J.GT.6) J=6
      T=TA(I)+TI(I)-TH(I)+((TH(I)+TF(I))/PKH(I,J))+((TM(I)+TF(I))/
     1PHM(I,J))*((1.0-PHH(I,J))/PKH(I,J)+PHH(I,J)-P(I,J))
      RETURN
      END
C
      SUBROUTINE SORT(I,M)
C
C   SUBROUTINE SORT IS USED TO SORT TARGETS IN ASCENDING RANGE
C   ORDER. THIS IS USED TO DETERMINE THE PRIORITY OF A TARGET
C   FOR FIRE ALLOCATION.
C
      CCMMON /GRP5/ LOT(6,6),ROT(6,6)
      DO 19 J=1,M
      IF(ROT(I,M).GE.ROT(I,J)) GOTO 19
   21 R=ROT(I,J)
      NN=LOT(I,J)
      RCT(I,J)=ROT(I,M)
      LOT(I,J)=LOT(I,M)
      RCT(I,M)=R
      LOT(I,M)=NN
   19 CONTINUE
      RETURN
      END
C
      SUBROUTINE KOVER(ZO,TMACT,SIZET,ZT,S,HTS,ZS,VISFRT)
C
C   SUBROUTINE KOVER DETERMINES WHAT PORTION OF A PARTICULA TARGET
C   IS COVERED BY THE TERRAIN BETWEEN THE TARGET AND OBSERVER.
C   THIS NUMBER IS USED IN THE DETECTION AND ATTRITION COMPUTATION.
C
      IF(S.EQ.0.) GO TO 2000
      IF(HTS.GE.ZS) GO TO 2050
      HEXT=ZO+(HTS-ZO)/S
      EVIST=AMAX1(HEXT,TMACT)
      IF(EVIST.GE.ZT) GO TO 2050
      IF(EVIST.LE.ZT-SIZET) RETURN
      VIS=(ZT-EVIST)/SIZET
      IF(VIS.LT.VISFRT) VISFRT=VIS
      RETURN
2000  IF(HTS.LT.ZO) RETURN
2050  VISFRT=0.0
      RETURN
      END
C
      SUBROUTINE LOS(XA,YA,TMACA,TMICA,SIZEA,XB,YB,TMACB,TMICB,SIZEB,
     -LATOB,LBTOA,VISFRA,VISFRB)
C
```

```
C   SUBROUTINE LOS COMPUTES A PERCENT OF A TARGET VISIBLE TO A          GRA16260
C   PARTICULAR OBSERVER,GIVEN THE COORDINATES OF BOTH.                   GRA16270
C                                                                       GRA16280
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100),SPRD(100)      GRA16290
      COMMON /HILLS/ ECC(100),PXX(100),PYY(100),PXY(100),BASE           GRA16300
      COMMON /HILLS/ NHILLS                                             GRA16310
      COMMON /COVER/ CXC(150),CYC(150),CPEAK(150),CPXX(150),CPYY(150)   GRA16320
      COMMON /COVER/ CPXY(150),NCVELS                                   GRA16330
      COMMON /COUNTR/KH, KHW, KV, KN, KGRS, KELL, KINT                  GRA16340
      COMMON /GRID/ LST(5,4),NHL(5,4),LISTH(150),KHREP(150),KTREP       GRA16350
      COMMON /GRID/ LSTC(5,4),NC(5,4),LISTC(400),KCREP(150)             GRA16360
      DIMENSION IGX(100),IGY(100),IEL(100),CS1(100),CS2(100)            GRA16370
      DATA GSIZE/1000./                                                 GRA16380
C SUBROUTINE TO COMPUTE FRACTION VISIBLE FOR OBSERVER TARGET PAIRS      GRA16390
      VISFRA=1.                                                         GRA16400
      VISFRB=1.                                                         GRA16410
      XBA=XB-XA                                                         GRA16420
      YBA=YB-YA                                                         GRA16430
      IF((XBA.EQ.0.).AND.(YBA.EQ.0.)) RETURN                            GRA16440
      IF(SIZEA+TMICA.LE.0.) GO TO 510                                   GRA16450
      IF(SIZEB+TMICB.LE.0.) GO TO 510                                   GRA16460
      IF(TMICA.LT.0.) VISFRA=1.0+TMICA/SIZEA                            GRA16470
      IF(TMICB.LT.0.) VISFRB=1.0+TMICB/SIZEB                            GRA16480
      ZA=TMACA + TMICA + SIZEA                                          GRA16490
      ZB=TMACB + TMICB + SIZEB                                          GRA16500
      KTREP=KTREP+1                                                     GRA16510
      ZBA=ZB-ZA                                                         GRA16520
      XBASQ=XBA*XBA                                                     GRA16530
      YBASQ=YBA*YBA                                                     GRA16540
      XYBA=XBA*YBA                                                      GRA16550
      TWOXBA=2.*XBA                                                     GRA16560
      TWOYBA=2.*YBA                                                     GRA16570
C COMPUTE GRID SQUARES CROSSED BY A TO B LINE                           GRA16580
      NGRSQ=0                                                           GRA16590
      IF(XBA) 110,95,100                                                GRA16600
95    XBA=0.1                                                           GRA16610
100   ISGX=-1                                                           GRA16620
      XINC=GSIZE/XBA                                                    GRA16630
      GO TO 120                                                         GRA16640
110   ISGX=1                                                            GRA16650
      XINC=-GSIZE/XBA                                                   GRA16660
120   IF(YBA)  140,125,130                                              GRA16670
125   YBA=0.1                                                           GRA16680
130   ISGY=-1                                                           GRA16690
      YINC=GSIZE/YBA                                                    GRA16700
      GO TO 150                                                         GRA16710
140   ISGY=1                                                            GRA16720
      YINC=-GSIZE/YBA                                                   GRA16730
```

```
150   IX=1+IFIX(XB/GSIZE)
      IY=1+IFIX(YB/GSIZE)
      XNEXT=GSIZE*(FLOAT(IX)+0.5*(ISGX-1.))
      YNEXT=GSIZE*(FLOAT(IY)+0.5*(ISGY-1.))
      XSTEP=(XB-XNEXT)/XBA
      YSTEP=(YB-YNEXT)/YBA
160   NGRSQ=NGRSQ+1
      IGX(NGRSQ)=IX
      IGY(NGRSQ)=IY
      IF((XSTEP.GT.1.).AND.(YSTEP.GT.1.)) GO TO 200
      IF(XSTEP-YSTEP) 170,180,190
170   IX=IX +ISGX
      XSTEP=XSTEP+XINC
      GO TO 160
180   IX=IX+ISGX
      XSTEP=XSTEP+XINC
190   IY=IY+ISGY
      YSTEP=YSTEP+YINC
      GO TO 160
200   KGRS=KGRS+NGRSQ
C GRID SQUARE LIST NOW COMPLETE IN IGX, IGY WITH NGRSQ ENTRIES
C
C NOW FIND WHICH COVER ELLIPSES TOUCH THE A TO B LINE,
C CHECK ELEVATIONS AT S1 AND S2 FOR EACH SUCH ELLIPSE
      NELS=0
      CHTMAX=0.
      IF(NCVELS.EQ.0) GOTO 270
      DO 260 K=1,NGRSQ
      IX=IGX(K)
      IY=IGY(K)
      N=NC(IX,IY)
      IF(N.EQ.0) GO TO 260
      LS=LSTC(IX,IY)
      LEND=LS+N-1
      DO 250 L=LS,LEND
      KELL=KELL+1
      IC=LISTC(L)
      IF(KCREP(IC).EQ.KTREP) GO TO 250
      KCREP(IC)=KTREP
      RX=XA-CXC(IC)
      RY=YA-CYC(IC)
      PPXX=CPXX(IC)
      PPYY=CPYY(IC)
      PPXY=CPXY(IC)
      AA=PPXX*XBASQ+PPYY*YBASQ+PPXY*XYBA
      BB=PPXX*TWOXBA*RX+PPYY*TWOYBA*RY+PPXY*(RX*YBA+RY*XBA)
      CC=PPXX*RX*RX+PPYY*RY*RY+PPXY*RX*RY-1.0
      ARG=BB*BB-4.0*AA*CC
```

```
      IF(ARG.LE.0.) GO TO 250                                           GRA17220
      SQ=SQRT(ARG)                                                      GRA17230
      S1=-(BB+SQ)/(2.0*AA)                                              GRA17240
      S2=(SQ-BB)/(2.0*AA)                                               GRA17250
      IF(S1.GE.1.) GO TO 250                                            GRA17260
      IF(S2.LE.0.) GO TO 250                                            GRA17270
      IF(S1.LE.0.)  GO TO 510                                           GRA17280
      IF(S2.GE.1.) GO TO 510                                            GRA17290
C NOW CHECK LOS AT S1 AND S2                                            GRA17300
      KINT=KINT+1                                                       GRA17310
      CPK=CPEAK(IC)                                                     GRA17320
      XS=XA+S2*XBA                                                      GRA17330
      YS=YA+S2*YBA                                                      GRA17340
      CALL ELEV(XS,YS,HTS)                                              GRA17350
      HTS=HTS+CPK                                                       GRA17360
      ZS=ZA+S2*ZBA                                                      GRA17370
      IF(LATOB.EQ.0) GO TO 210                                          GRA17380
      CALL KOVER(ZA,TMACB,SIZEB,ZB,S2,HTS,ZS,VISFRB)                    GRA17390
      IF(VISFRB.LE.0.) GO TO 510                                        GRA17400
210   IF(LBTOA.EQ.0) GO TO 220                                          GRA17410
      S=1.-S2                                                           GRA17420
      CALL KOVER(ZB,TMACA,SIZEA,ZA,S,HTS,ZS,VISFRA)                     GRA17430
      IF(VISFRA.LE.0.) GO TO 510                                        GRA17440
220   XS=XA+S1*XBA                                                      GRA17450
      YS=YA+S1*YBA                                                      GRA17460
      CALL ELEV(XS,YS,HTS)                                              GRA17470
      HTS=HTS+CPK                                                       GRA17480
      ZS=ZA+S1*ZBA                                                      GRA17490
      IF(LATOB.EQ.0) GO TO 230                                          GRA17500
      CALL KOVER(ZA,TMACB,SIZEB,ZB,S1,HTS,ZS,VISFRB)                    GRA17510
      IF(VISFRB.LE.0.) GO TO 510                                        GRA17520
230   IF(LBTOA.EQ.0) GO TO 240                                          GRA17530
      S=1.0-S1                                                          GRA17540
      CALL KOVER(ZB,TMACA,SIZEA,ZA,S,HTS,ZS,VISFRA)                     GRA17550
      IF(VISFRA.LE.0.) GO TO 510                                        GRA17560
240   NELS=NELS+1                                                       GRA17570
      IEL(NELS)=IC                                                      GRA17580
      CS1(NELS)=S1                                                      GRA17590
      CS2(NELS)=S2                                                      GRA17600
      IF(CPK.GT.CHTMAX) CHTMAX=CPK                                      GRA17610
250   CONTINUE                                                          GRA17620
260   CONTINUE                                                          GRA17630
C ALL ELLIPSES CHECKED                                                  GRA17640
C                                                                       GRA17650
C NOW START ON THE HILLS                                                GRA17660
  270 DO 600 K=1,NGRSQ                                                  GRA17670
      IX=IGX(K)                                                         GRA17680
      IY=IGY(K)                                                         GRA17690
```

```
      IF(NHL(IX,IY).EQ.0) GO TO 600
      LS=LST(IX,IY)
      LEND=LS+NHL(IX,IY) -1
      DO 500 L=LS,LEND
      I=LISTH(L)
      IF(KHREP(I).EQ.KTREP) GO TO 500
      KHREP(I)=KTREP
C  PROCESSING FOR HILL I STARTS HERE
      KH=KH+1
C COMPUTE W =TOP OF THIS HILL ALONG O-T LINE
C
      TRX=XA-XC(I)
      TRY=YA-YC(I)
      TPXX=PXX(I)
      TPYY=PYY(I)
      TPXY=PXY(I)
      FQ=TWOXBA*TPXX*TRX+TWOYBA*TPYY*TRY+TPXY*(TRX*YBA+TRY*XBA)
      GQ=TPXX*XBASQ+TPYY*YBASQ+TPXY*XYBA
      IF(GQ .EQ.0.)  GO TO 500
      W=-FQ/(2.*GQ)
      IF(ABS(W).GT.5.) GO TO 500
      FSQ=FQ*FQ
      EQ=TPXX*TRX*TRX+TPYY*TRY*TRY+TPXY*TRX*TRY
C
      POWER=EQ-FSQ/(4.*GQ)
      IF(POWER .LT. -3.) GO TO 500
      HHW=PEAK(I)*EXP(POWER)
      KHW=KHW+1
      IF(HHW.LE.BASE) GO TO 500
      ZW=ZA+W*ZBA
      IF((W.LT.0.).OR.(W.GT.1.)) GO TO 300
      IF(HHW .GE. ZW) GO TO 510
      CVHTW=0.
      IF(NELS.EQ.0) GO TO 300
      DO 280 M=1,NELS
      IF((CS1(M).GE.W).OR.(CS2(M).LE.W)) GO TO 280
      IC=IEL(M)
      IF(CVHTW.LT.CPEAK(IC))  CVHTW=CPEAK(IC)
280   CONTINUE
      IF((HHW+CVHTW).GE.ZW)  GO TO 510
300   IF(HHW+CHTMAX.LT.AMIN1(ZA-SIZEA,ZB-SIZEB)) GO TO 500
C IF WE GET TO HERE THEN NEED TO FIND LOWEST SIGHT LINE OVER HILL
C NEWTON ITERATION A TO B GIVING VISFRB
      IF(LATOB.EQ.0) GO TO 400
      KV=KV+1
      V=W
      HHV=HHW
      NCT=0
```

```
      FV=FQ*V                                                           GRA18180
      TWOGV=2.*GQ*V                                                     GRA18190
330    FCNV=ZA+HHV*(TWOGV*V+FV-1.)                                      GRA18200
      KN=KN+1                                                           GRA18210
      FACTOR=(TWOGV*TWOGV+2.*(GQ+TWOGV*FQ)+FSQ)                         GRA18220
      DFCNV=HHV*V*FACTOR                                                GRA18230
      IF(ABS(DFCNV) .LT.1.E-10) GO TO 350                               GRA18240
      V=V-FCNV/DFCNV                                                    GRA18250
      FV=FQ*V                                                           GRA18260
      TWOGV=2.*GQ*V                                                     GRA18270
      POWER =  EQ+FV+GQ*V*V                                             GRA18280
      IF(POWER .LT. -3.) GO TO 400                                      GRA18290
      HHV=PEAK(I)*EXP(POWER)                                            GRA18300
      DHHV=HHV*(FQ+TWOGV)                                               GRA18310
      ELV=ZA+DHHV*V                                                     GRA18320
      IF (ABS(HHV-ELV) .LT. 1.) GO TO 350                               GRA18330
      NCT=NCT+1                                                         GRA18340
      IF (NCT.LT.10) GO TO 330                                          GRA18350
350   IF((V.LT.0.).OR.(V.GT.1.)) GO TO 400                              GRA18360
      CVHTV=0.                                                          GRA18370
      IF(NELS.EQ.0) GO TO 390                                           GRA18380
      DO 380 M=1,NELS                                                   GRA18390
      IF((CS1(M).GE.V).OR.(CS2(M).LE.V))GO TO 380                       GRA18400
      IC=IEL(M)                                                         GRA18410
      IF(CVHTV.LT.CPEAK(IC)) CVHTV=CPEAK(IC)                            GRA18420
380   CONTINUE                                                          GRA18430
390   HTV=HHV+CVHTV                                                     GRA18440
      ZV=ZA+V*ZBA                                                       GRA18450
      CALL KOVER(ZA,TMACB,SIZEB,ZB,V,HTV,ZV,VISFRB)                     GRA18460
      IF (VISFRB.LE.0.) GO TO 510                                       GRA18470
C NEWTON ITERATION B TO A GIVING VISFRA                                 GRA18480
      IF(ABS(V).GT.5.)GO TO 400                                         GRA18490
400   IF(LBTOA.EQ.0) GO TO 500                                          GRA18500
      KV=KV+1                                                           GRA18510
      V=W                                                               GRA18520
      VM1=V-1.                                                          GRA18530
      HHV=HHW                                                           GRA18540
      NCT=0                                                             GRA18550
      FV=FQ*V                                                           GRA18560
      TWOGV=2.*GQ*V                                                     GRA18570
430   FCNV=ZB+HHV*((FQ+TWOGV)*VM1-1.)                                   GRA18580
      KN=KN+1                                                           GRA18590
      FACTOR=(TWOGV*TWOGV+2.*(GQ+TWOGV*FQ)+FSQ)                         GRA18600
      DFCNV=HHV*VM1*FACTOR                                              GRA18610
      IF(ABS(DFCNV) .LT.1.E-10) GO TO 450                               GRA18620
      V=V-FCNV/DFCNV                                                    GRA18630
      IF(ABS(V).GT.5.)GO TO 500                                         GRA18640
      VM1=V-1.                                                          GRA18650
```

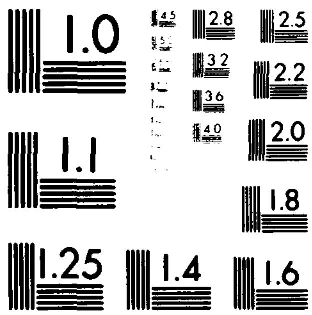
```
      FV=FQ*V
      TWOGV=2.*GQ*V
      POWER =  EQ+FV+GQ*V*V
      IF(POWER .LT. -3.) GO TO 500
      HHV=PEAK(I)*EXP(POWER)
      DHHV=HHV*(FQ+TWOGV)
      ELV=ZB+DHHV*VM1
      IF (ABS(HHV-ELV) .LT. 1.) GO TO 450
      NCT=NCT+1
      IF (NCT.LT.10) GO TO 430
  450 IF((V.LT.0.).OR.(V.GT.1.)) GO TO 500
      CVHTV=0.
      IF(NELS.EQ.0) GO TO 490
      DO 480 M=1,NELS
      IF((CS1(M).GE.V).OR.(CS2(M).LE.V))GO TO 480
      IC=IEL(M)
      IF(CVHTV.LT.CPEAK(IC)) CVHTV=CPEAK(IC)
  480 CONTINUE
  490 HTV=HHV+CVHTV
      ZV=ZA+V*ZBA
      S=-VM1
      CALL KOVER(ZB,TM,CA,SIZEA,ZA,S,HTV,ZV,VISFRA)
      IF (VISFRA.LE.0.) GO TO 510
  500 CONTINUE
  600 CONTINUE
      RETURN
  510 VISFRA=0.
      VISFRB=0.
      RETURN
      END
```

AD-A109 689 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 15/7
AN OPERATIONAL LANCHESTER-TYPE MODEL OF SMALL-UNIT AMPHIBIOUS O--ETC(U)
SEP 81 S D PARK

UNCLASSIFIED NL

2 OF 2

END
DATE
FILMED
82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX E

PLOTTING PROGRAM FOR TERRAIN CONTOUR LINE

```
// EXEC FRTXCLGP
//FORT.SYSIN DD *
      IMPLICIT REAL*4(A-H,O-Z)
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100)
      COMMON /HILLS/ SPRD(100),ECC(100),PXX(100),PYY(100)
      COMMON /HILLS/ PXY(100),BASE,NHILLS
      COMMON /GRID/ LST(5,4),NHL(5,4),LISTH(150),KHREP(100)
      COMMON /GRID/ KTREP
      DIMENSION XXH(25000),YYH(25000)
C
C    PROGRAM READ IN HILL MASS DATA AND CALCULATE VALUE OF
C    VARIABLE FOR THE ELEVATION SUBROUTINE
C    THE DEFINITION OF DATA-IN VARIABLES SHOULD BE REFERED
C    TO(REF.5)
C
      PAI=3.14159
      L = 5
      READ(L,7) NHILLS
      READ(L,47) BASE
47    FORMAT(F10.4)
7     FORMAT(I6)
17    FORMAT(6F10.3)
      DO 50 I=1,NHILLS
      READ(L,17) XC(I),YC(I),PEAK(I),ANGH(I),SPRD(I),ECC(I)
50    CONTINUE
      READ(L,37)LST
      READ(L,37)NHL
      READ(L,7)NHTOT
      READ(L,37)(LISTH(I),I=1,NHTOT)
37    FORMAT(10I5)
   65 DO 100 I=1,NHILLS
      ANGLE=ANGH(I)*PAI/180.
      SANG=SIN(ANGLE)
      CANG=COS(ANGLE)
      A=PEAK(I)/(PEAK(I)-50.)
      A=ALOG(A)
      B=A*ECC(I)**2
      SPD=SPRD(I)**2
      PXX(I)=-(A*CANG*CANG+B*SANG*SANG)/SPD
      PYY(I)=-(A*SANG*SANG+B*CANG*CANG)/SPD
      PXY(I)=(2.*SANG*CANG*(B-A))/SPD
      KHREP(I)=-2147483600
C  ALL VALUES NOW IN METERS ON 0 -- 10,000 GRID
100   CONTINUE
      KTREP=-2147483600
C
C    MAX AND MIN COORDINATES OF THE REAL TERRAIN REQUIRED
C    FOR THE PLOTTER ROUTINE. IN THIS CASE 3X4 KM TERRAIN
C    WILL BE PLOTTED
C
      XMAX=5000.
      YMAX=4000.
      XMIN=1000.
      YMIN=1000.
      XXH(1)=1000.
      YYH(1)=1000.
      XXH(2)=1000.
      YYH(2)=4000.
      XXH(3)=5000.
      YYH(3)=4000.
```

```
      XXH(4)=5000.
      YYH(4)=1000.
C
C   USING THE BISECTION SEARCH METHOD, FIND THE LOCATION
C   WHICH HAS A ELEVATION,I,E. 20, 40,..ETC.
C
      ME=5
      YELTA=20.
      XELTA=5.
      XEL=1000.
      YEL=1000.
      YCON=1000
      XEND=5000
      YEND=4000.
  991 CALL ELEV(XEL,YEL,ZNEW)
      TELTA=YELTA
      NVAL1=IFIX(ZNEW/20.)
  717 ZOLD=ZNEW
  727 YEL=YEL+YELTA
      IF(YEL.LT.YEND) GO TO 777
      XEL=XEL+5.
      IF(XEL.EQ.XEND) GO TO 333
      YEL=YCON
      GO TO 991
  777 CALL ELEV(XEL,YEL,ZNEW)
      NVAL2=IFIX(ZNEW/20.)
      IZZ=IABS(NVAL2-NVAL1)
      IF(IZZ.GE.1) GO TO 788
      NVAL1=NVAL2
      GO TO 717
  788 KVAL=NVAL2
      YRES=YEL
      IF(NVAL2.GT.NVAL1) GO TO 630
      ZVAL=NVAL1*20.
      NK=1
      TELTA=YELTA
  631 YEL=YEL-TELTA/2.
  632 CALL ELEV(XEL,YEL,ZNEW)
      IF(ZNEW.LT.ZVAL+0.1.AND.ZNEW.GT.ZVAL-0.1) GO TO 800
      IF(NK.GT.4) GO TO 800
      ZOLD=ZNEW
      TELTA=TELTA/2.
      NK=NK+1
      IF(ZNEW.LT.ZVAL) GO TO 631
      YEL=YEL+TELTA/2.0
      GO TO 632
C
C  CALCULATE INCREASING ELEVATION
C
  630 ZVAL=NVAL2*20.
      NK=1
      TELTA=YELTA
  642 YEL=YEL-TELTA/2.0
  643 CALL ELEV(XEL,YEL,ZNEW)
      IF(ZNEW.LT.ZVAL+0.1.AND.ZNEW.GT.ZVAL-0.1) GO TO 800
      IF(NK.GT.4) GO TO 800
      ZOLD=ZNEW
      TELTA=TELTA/2.0
      NK=NK+1
      IF(ZNEW.GT.ZVAL) GO TO 642
      YEL=YEL+TELTA/2.0
      GO TO 643
C
C  COLLECT ELEVATION COORINATE DATA WHICH IS WANTED TO BE
C  PLOTTED.
C
  800 IF(ZVAL.LT.20.) GO TO 189
```

```
      IF(ZVAL.GT.140.) GO TO 189
      IF(ZVAL.NE.20.AND.ZVAL.LT.100.) GO TO 189
      XXH(ME)=XEL
      YYH(ME)=YEL
      ME=ME+1
  189 NVAL1=KVAL
      YEL=YRES
      GO TO 727
  333 NP=ME-1
      CALL PLOTG(XXH,YYH,NP,1,0,75,'X-AXIS LABEL',12,
     1           'Y-AXIS LABEL',12,XMIN,XMAX,YMIN,YMAX,8.,6.)
      CALL PLOT(0.,0.,999)
      STOP
      END
      SUBROUTINE ELEV(X,Y,TMAC)
C
C   COMPUTE THE ELEVATION FOR A GIVEN X,Y COORDINATE
C
      IMPLICIT REAL*4(A-H,O-Z)
      COMMON /HILLS/ XC(100),YC(100),PEAK(100),ANGH(100)
      COMMON /HILLS/ SPRD(100),ECC(100),PXX(100),PYY(100)
      COMMON /HILLS/ PXY(100),BASE,NHILLS
      COMMON /GRID/ LST(5,4),NHL(5,4),LISTH(100),KHREP(100)
      COMMON /GRID/ KTREP
      DATA GSIZE/1000./
      ZMAX=BASE
      IX=1+IFIX(X/GSIZE)
      IY=1+IFIX(Y/GSIZE)
      IF(NHL(IX,IY).EQ.0) GO TO 150
      LS=LST(IX,IY)
      LEND=LS+NHL(IX,IY)-1
      DO 100 L=LS,LEND
      I=LISTH(L)
      QX=X-XC(I)
      QY=Y-YC(I)
      QXSQ=QX*QX
      QYSQ=QY*QY
      QXY=QX*QY
      FACTOR=PXX(I)*QXSQ+PYY(I)*QYSQ+PXY(I)*QXY
      IF(FACTOR.LT.-3.) GO TO 100
      HT=PEAK(I)*EXP(FACTOR)
      IF(HT.LE.ZMAX) GO TO 100
      ZMAX=HT
  100 CONTINUE
  150 TMAC=ZMAX
      RETURN
      END
```

LIST OF REFERENCES

1. Bonder, S., "The Lanchester Attrition-Rate Coefficient," Operations Research, v. 15, p. 221-232, May 1967.

2. Bonder, S., "An Overview of Land Battle Modeling in the U.S." In: Proceedings 13th U.S. Army Operations Research Symposium, pp. 73-77, 1974.

3. Chardwick, D.L., The Evaluation of Design and Employment Alternative for the LVA: A Modelling Strategy, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1978.

4. Farmer, W.T., A Survey of Approaches to the Modeling of Land Combat, M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1980.

5. Hartman, J.K., Parametric Terrain and Line-of-Sight Modeling in the STAR Combat Model, Naval Postgraduate School, Monterey, California, 1979.

6. Mills, G.M., The Enrichment of Smoller's Model of Land Combat, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1980.

7. Morris, W.T., "On the Art of Modeling," Management Science 13, B707-B717, August 1967.

8. Needels, C.G., Parameterization of Terrain in Army Combat Models, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1976.

9. Smoller, J., Operational Lanchester-type Model of Small Unit Land Combat, M.S. Thesis, Naval Postgraduate School, Monterey, California, September 1979.

10. Taylor, J.G., Force-on-Force Attrition Modeling, Military Application Section of Operations Research of America, January 1980.

11. Wallace, W.S., and Hagewood, E.G., Simulation of Tactical Alternative Responses (STAR), M.S. Thesis, Naval Postgraduate School, Monterey, California, December 1978.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Professor James G. Taylor, Code 55Tw Department of Operations Research Naval Postgraduate School Monterey, California 93940	12
5.	Major Jeffrey L. Elles, Code 55EI Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	Operations Research Department Deputy Chief of Staff for Strategy and Plans ROKA HQ, Seoul, Korea	1
7.	LTC Park Soon Dae Office of Operations Research/System Analysis ROKN HQ, Seoul, Korea	2

FILMED

2-8